(12) UK Patent Application (19) GB (11) 2 295 738 (13) A

(43) Date of A Publication 05.06.1996

(21) Application No 9522616.3

(22) Date of Filing 03.11.1995

(30) Priority Data
(31) 08334879 (32) 04.11.1994 (33) US

(71) Applicant(s)
Altera Corporation

(Incorporated in USA - California)

2610 Orchard Parkway, San Jose,
California 95134-2020, United States of America

(72) Inventor(s)
Bruce B Pedersen
Richard G Cliff

(74) Agent and/or Address for Service
Boult Wade Tennant
27 Furnival Street, LONDON, EC4A 1PQ,
United Kingdom

(51) INT CL^6
H03K 19/177

(52) UK CL (Edition O)
H3P PHX
G4H HU H13D
H3T T1H T2B2 T2B3

(56) Documents Cited
WO 94/17595 A1 US 5350954 A US 5220214 A

(58) Field of Search
UK CL (Edition O) G4H HU, H3P PHX
INT CL^6 H03K 19/173 19/177

(54) Programmable logic blocks with enhanced flexibility

(57) Flexibility in a programmable logic device is improved by allowing adjacent logic elements 12 in a logic array block to be cascaded selectively via AND gates 94. Further, an input carry signal 70a may be selected from either 70b an adjacent logic array block or 71a any other block via the local and global interconnections 24, 20, 22. One or more output signal lines (55, figure 5) may be connected to device output pins directly from the logic array block's internal interconnection elements 24, 26. The logic elements may comprise partitioned look-up tables (figures 9 and 10). The logic blocks may comprise a wide-input AND gate operating on the logic element outputs (figure 11). The logic elements may be arranged in two levels within the logic blocks (figure 16), the second level logic elements operating on the outputs from the first level logic elements and inputs to the logic block.

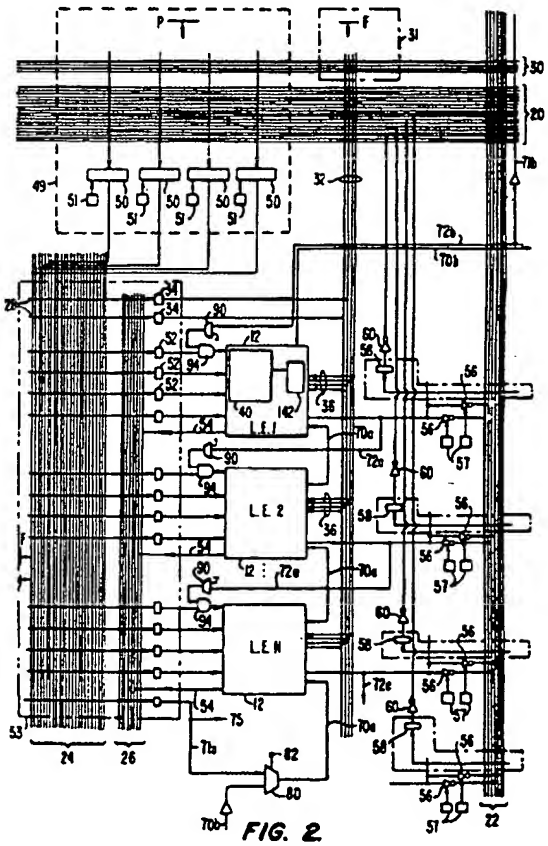

FIG. 2.

At least one drawing originally filed was informal and the print reproduced here is taken from a later filed formal copy.

This print takes account of replacement documents submitted after the date of filing to enable the application to comply with the formal requirements of the Patents Rules 1995

GB 2 295 738 A

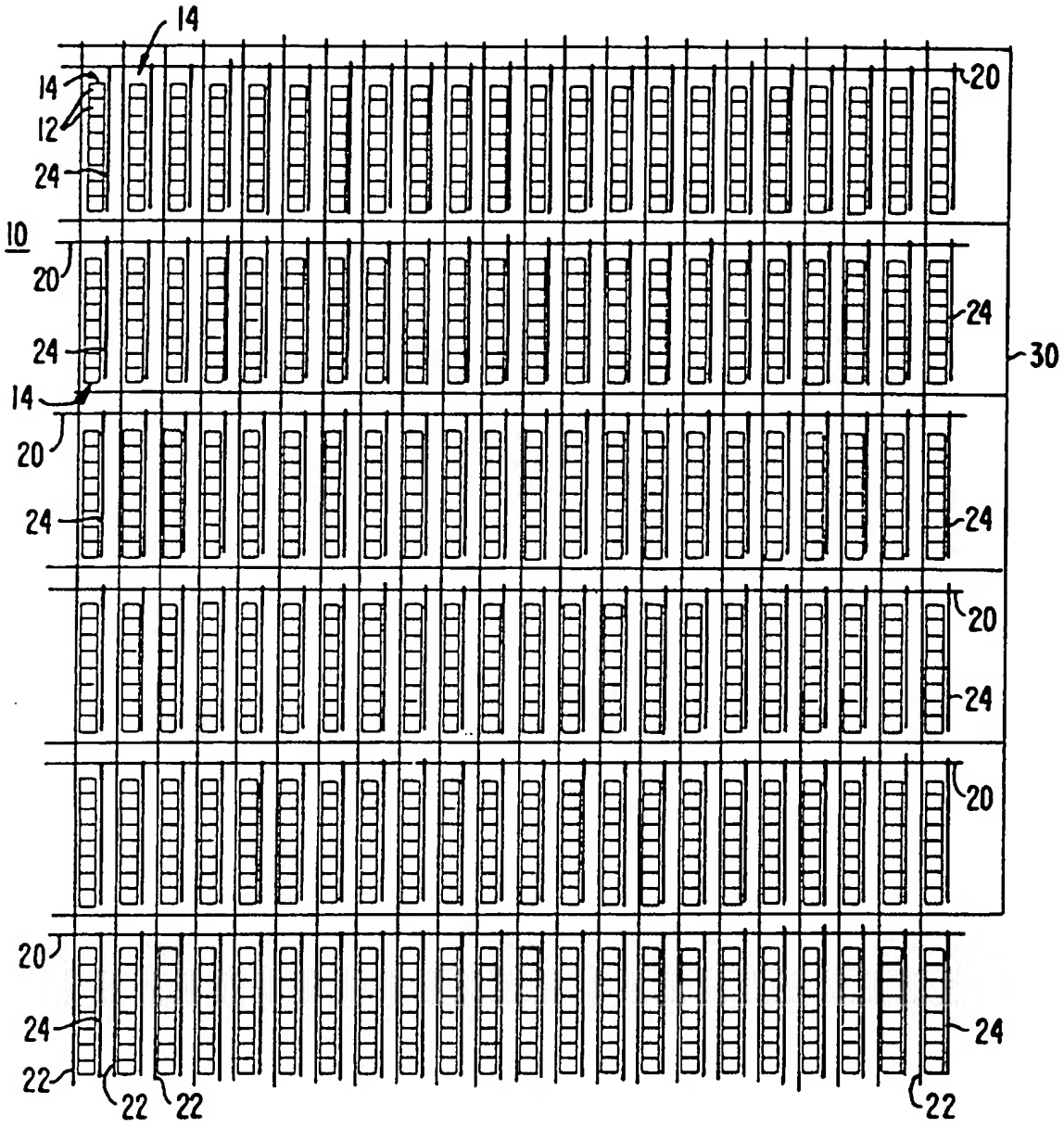

FIG. 1.

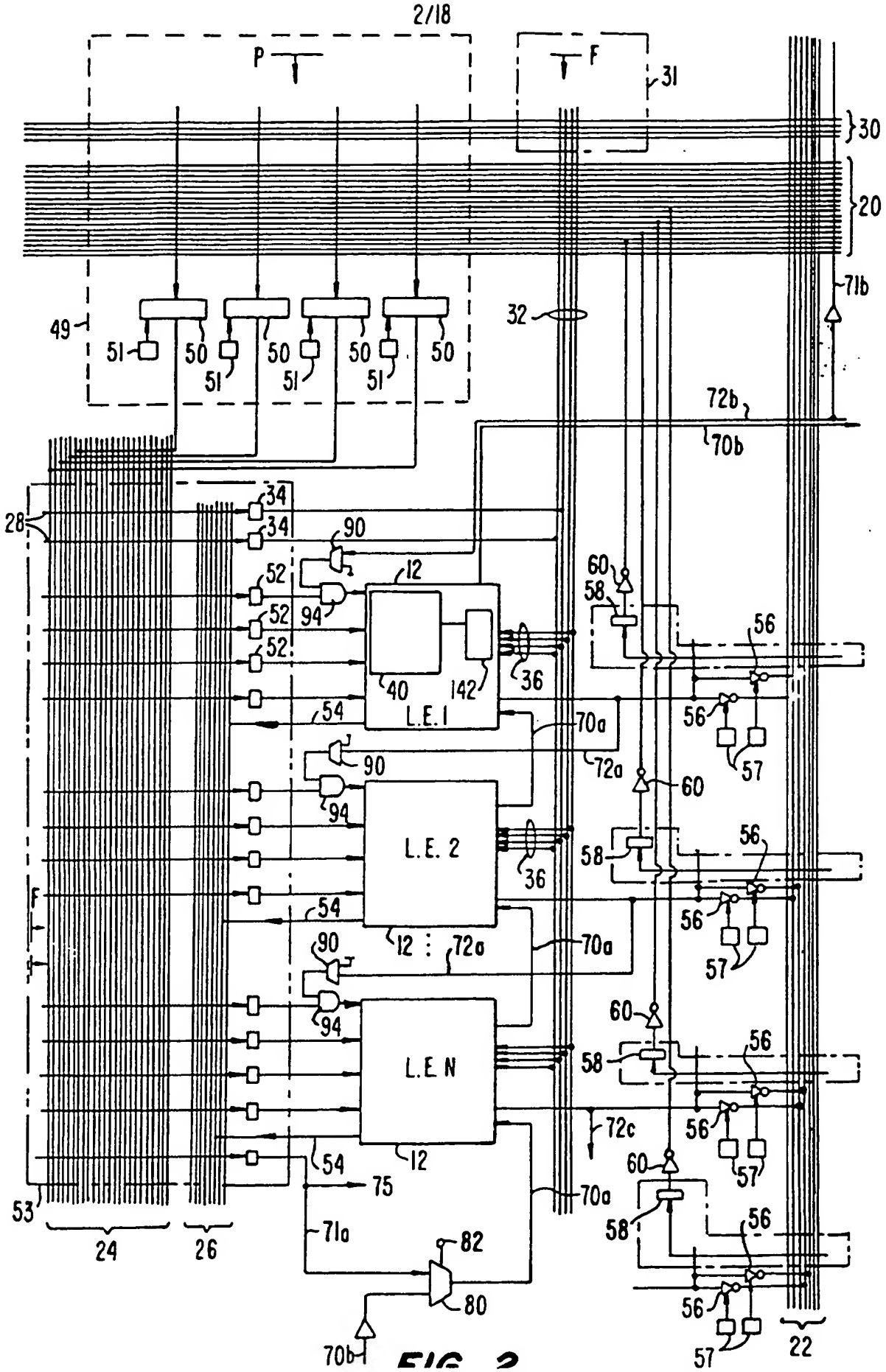
P
F
31
30
20
71b
49
51
50
32
72b
70b
28
34
90
52
94
12
L.E.1
40
142
36
54
70a
72a
60
58
56
57
L.E. 2
L.E N
F
53
24
26
75
71a
82
80
70b
72c
22

FIG. 2

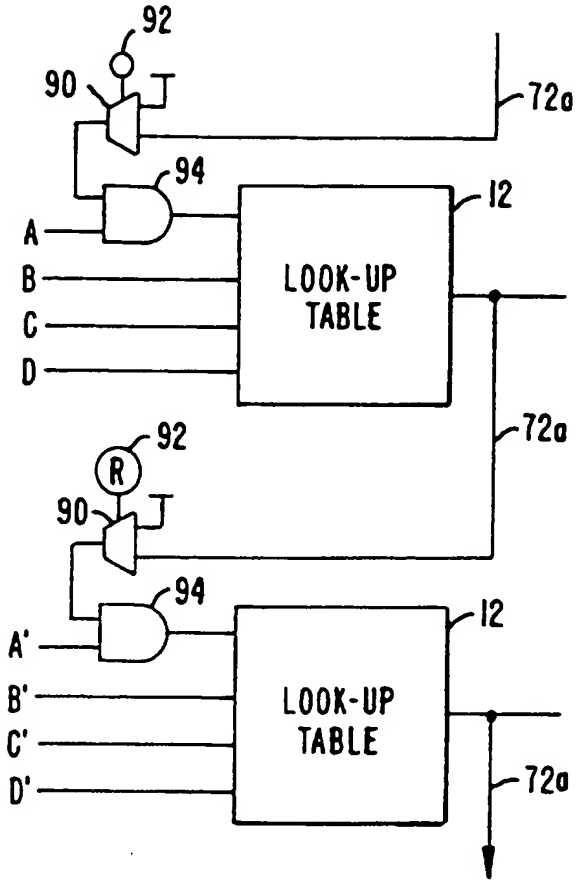
92
90
72a
94
12
A
B
C
D
LOOK-UP TABLE
72a
92
R
90
94
12
A'
B'
C'
D'
LOOK-UP TABLE
72a

FIG. 3.

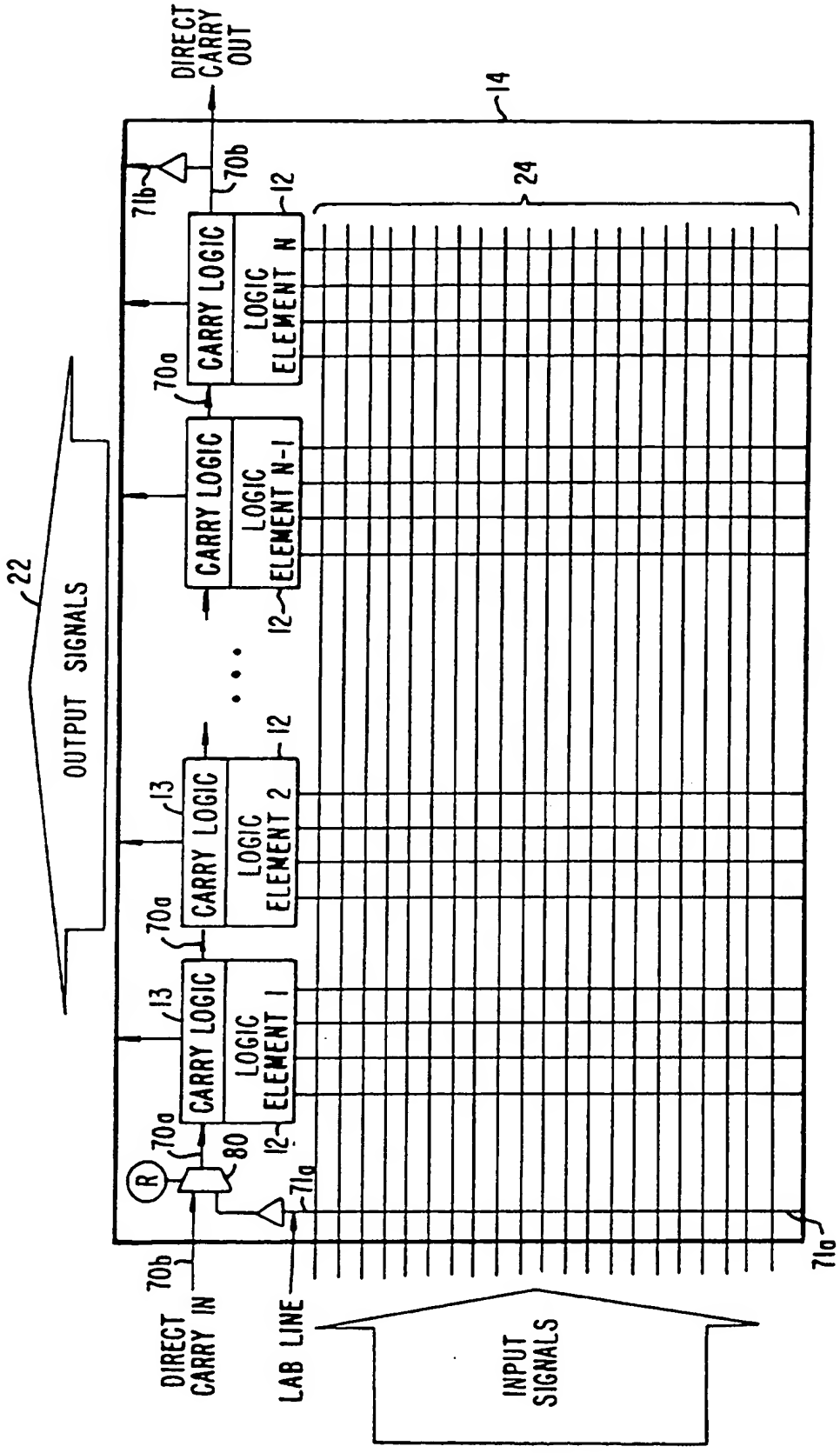
DIRECT CARRY OUT
14
24
22
OUTPUT SIGNALS
CARRY LOGIC
LOGIC ELEMENT N
LOGIC ELEMENT N-1
LOGIC ELEMENT 2
LOGIC ELEMENT 1
70a
70b
71a
71b
12
13
80
R
DIRECT CARRY IN
LAB LINE
INPUT SIGNALS

FIG. 4.

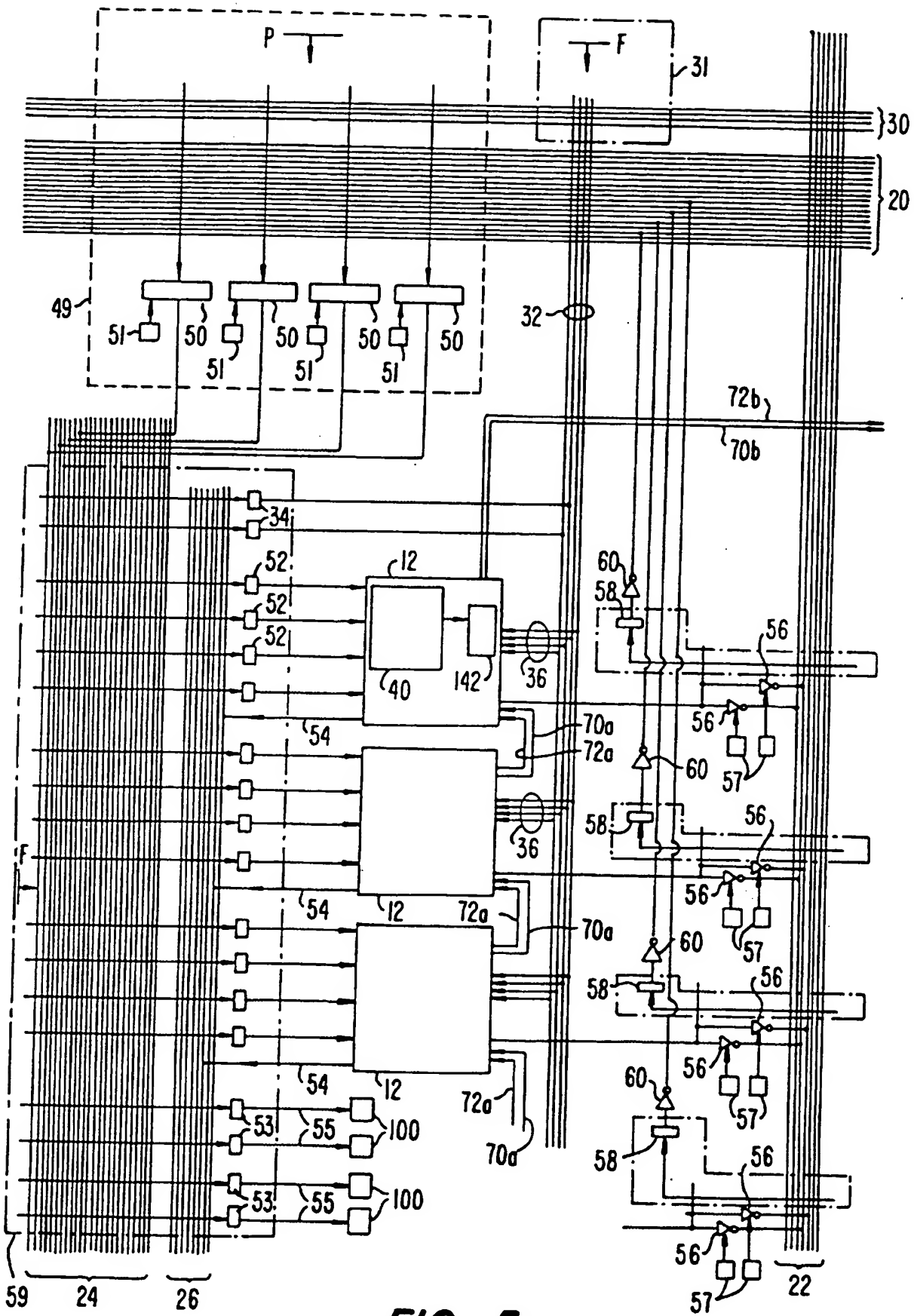
P
F
31
30
20
49
51
50
32
72b
70b
34
52
12
40
142
36
54
70a
72a
60
58
56
57
53
55
100
59
24
26
22

FIG. 5

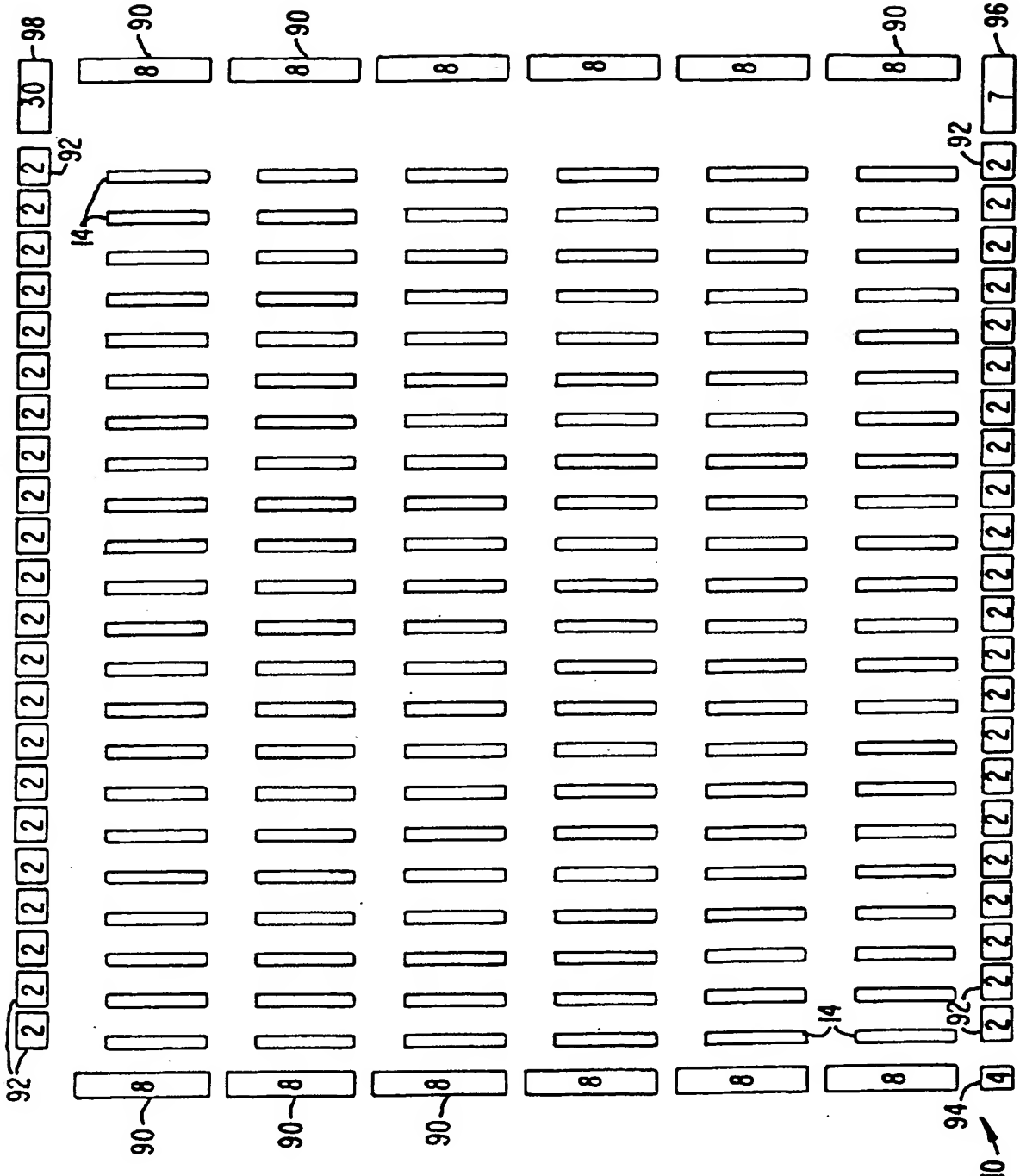

FIG. 6.

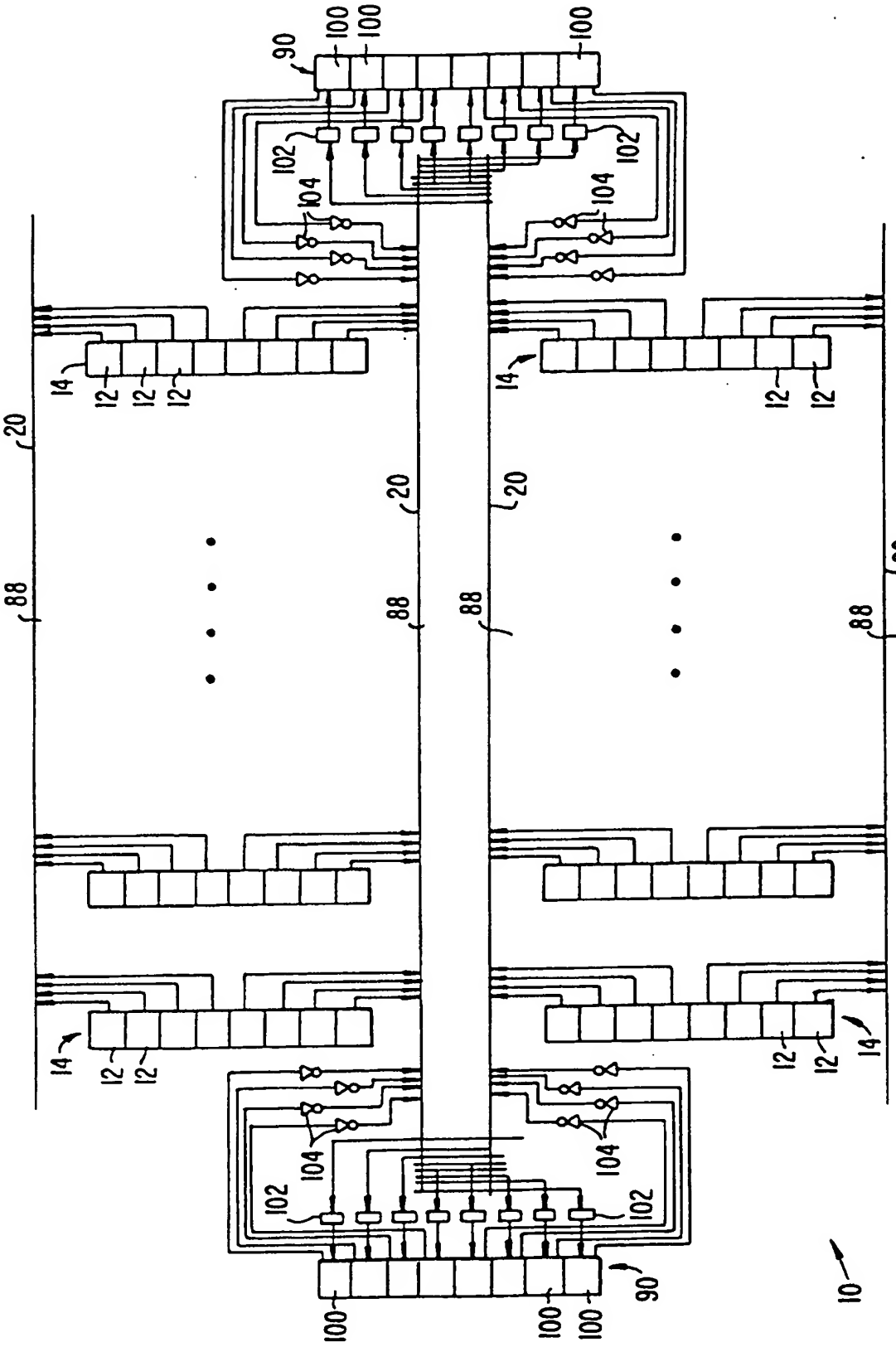

FIG. 7.

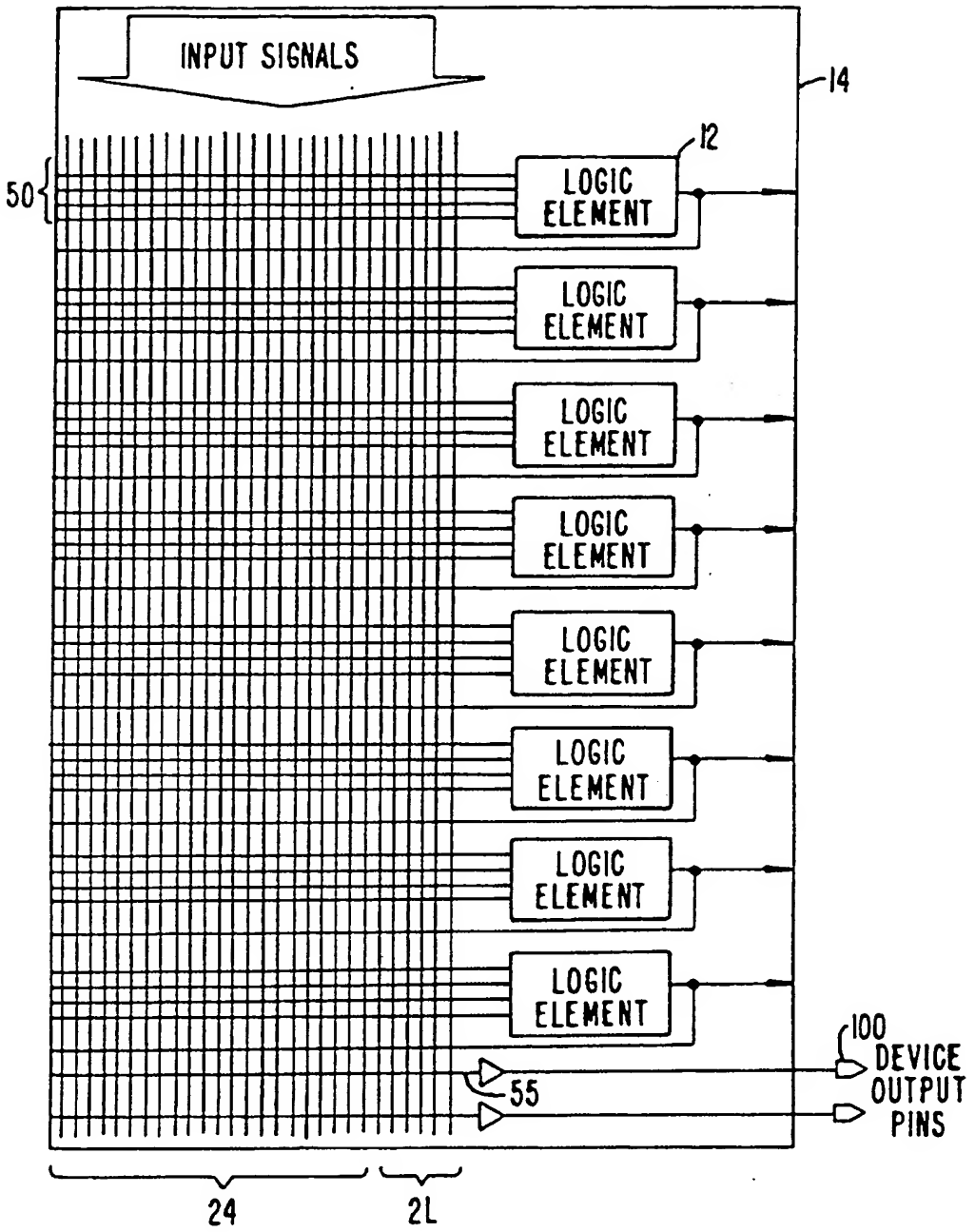
INPUT SIGNALS
14
12
50
LOGIC ELEMENT
LOGIC ELEMENT
LOGIC ELEMENT
LOGIC ELEMENT
LOGIC ELEMENT
LOGIC ELEMENT
LOGIC ELEMENT
LOGIC ELEMENT
100
DEVICE OUTPUT PINS
55
24
2L

FIG. 8.

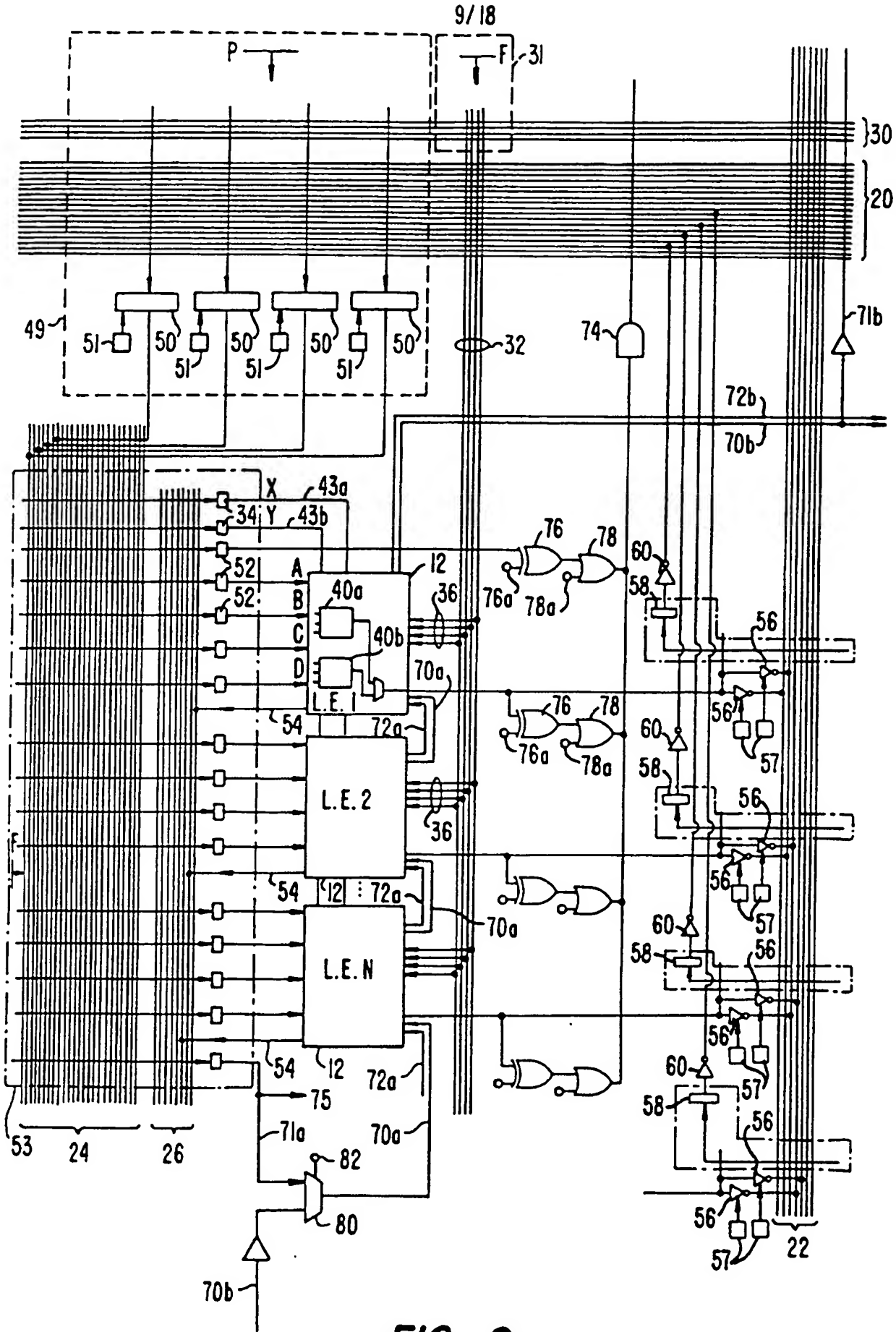
P
F
31
30
20
49
51
50
32
74
71b
72b
70b
X
43a
34
Y
43b
52
A
B
C
D
12
40a
40b
36
70a
L.E. 1
54
72a
L.E. 2
L.E. N
76
78
76a
78a
60
58
56
57
75
71a
82
80
53
24
26
22

FIG. 9

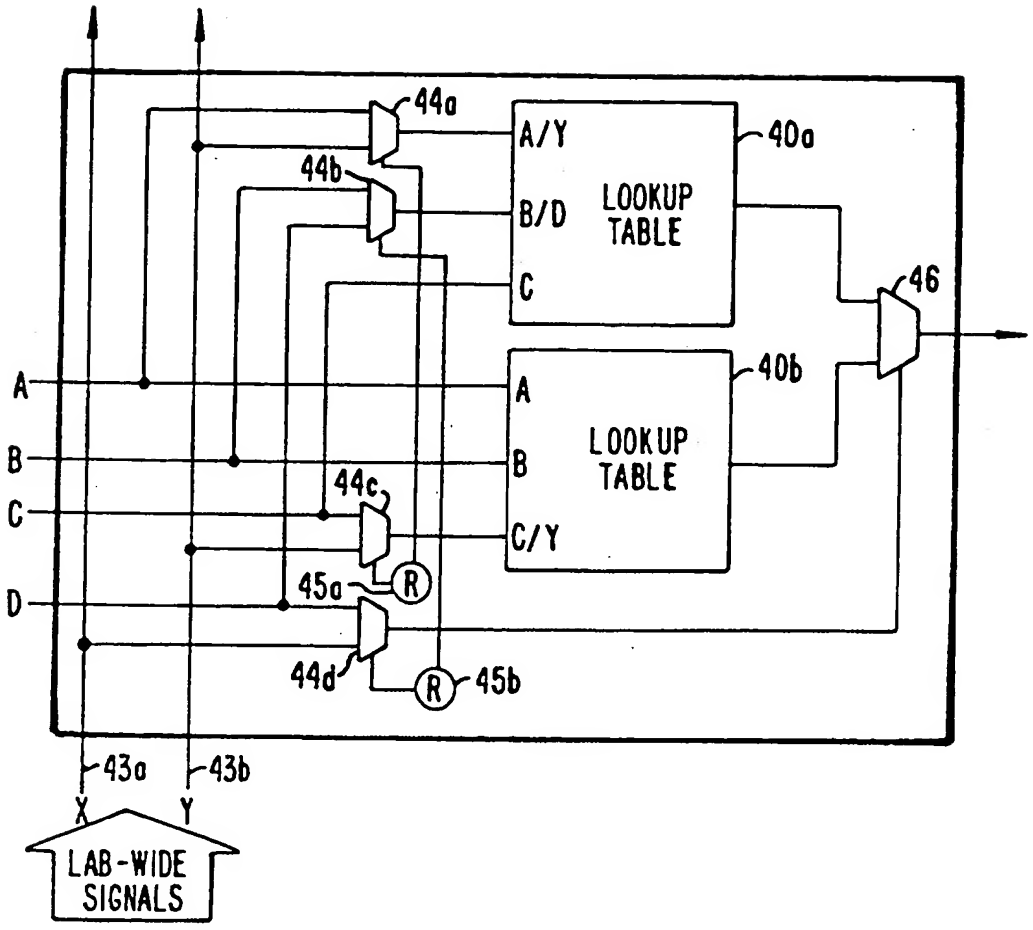
44a
44b
44c
44d
45a
45b
A/Y
B/D
C
LOOKUP TABLE
40a
A
B
C/Y
LOOKUP TABLE
40b
46
A
B
C
D
R
R
43a
43b
X
Y
LAB-WIDE SIGNALS

FIG. 10.

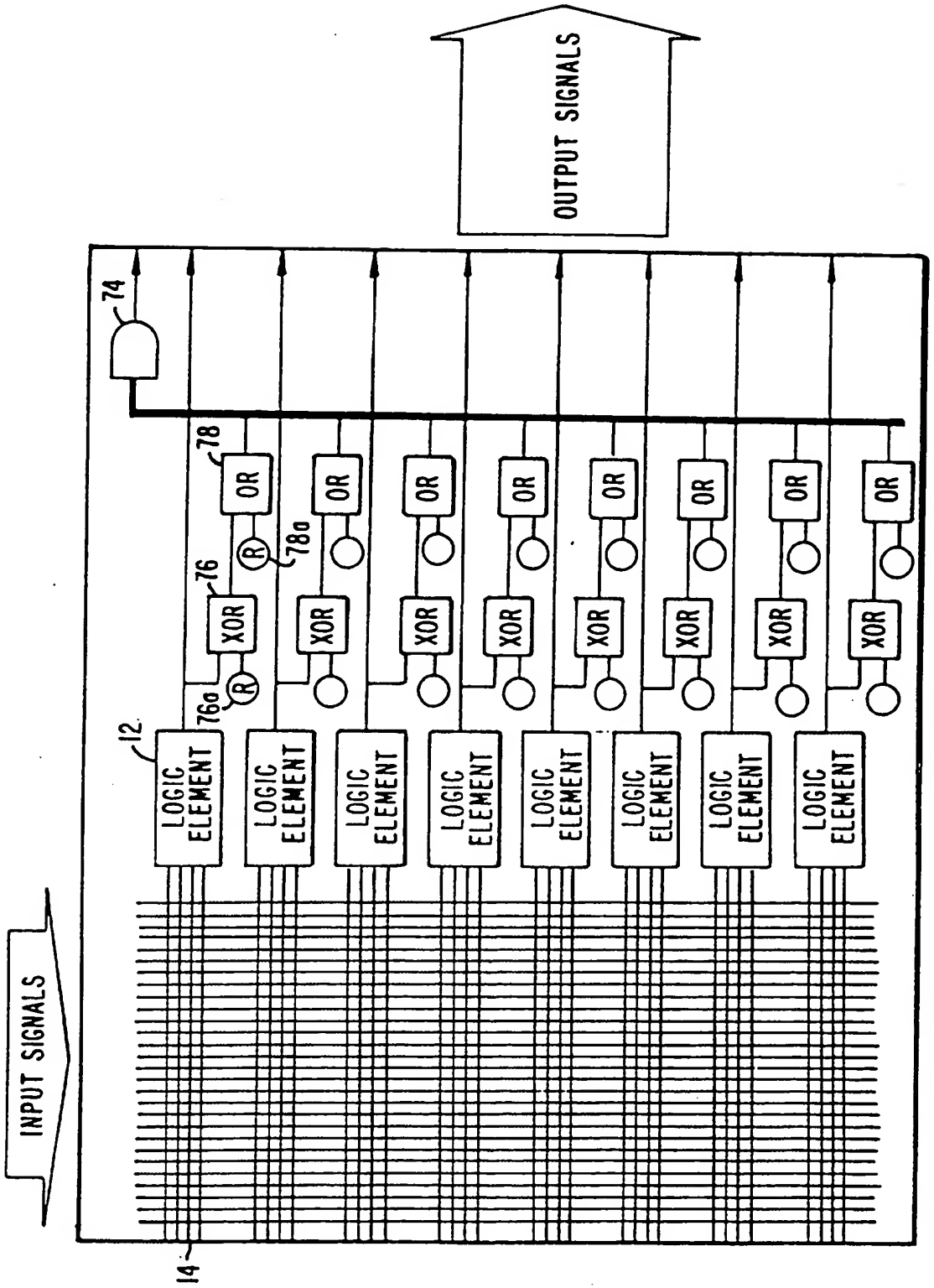
OUTPUT SIGNALS
INPUT SIGNALS
74
78
78a
76
76a
OR
XOR
LOGIC ELEMENT
12
14

FIG. 11.

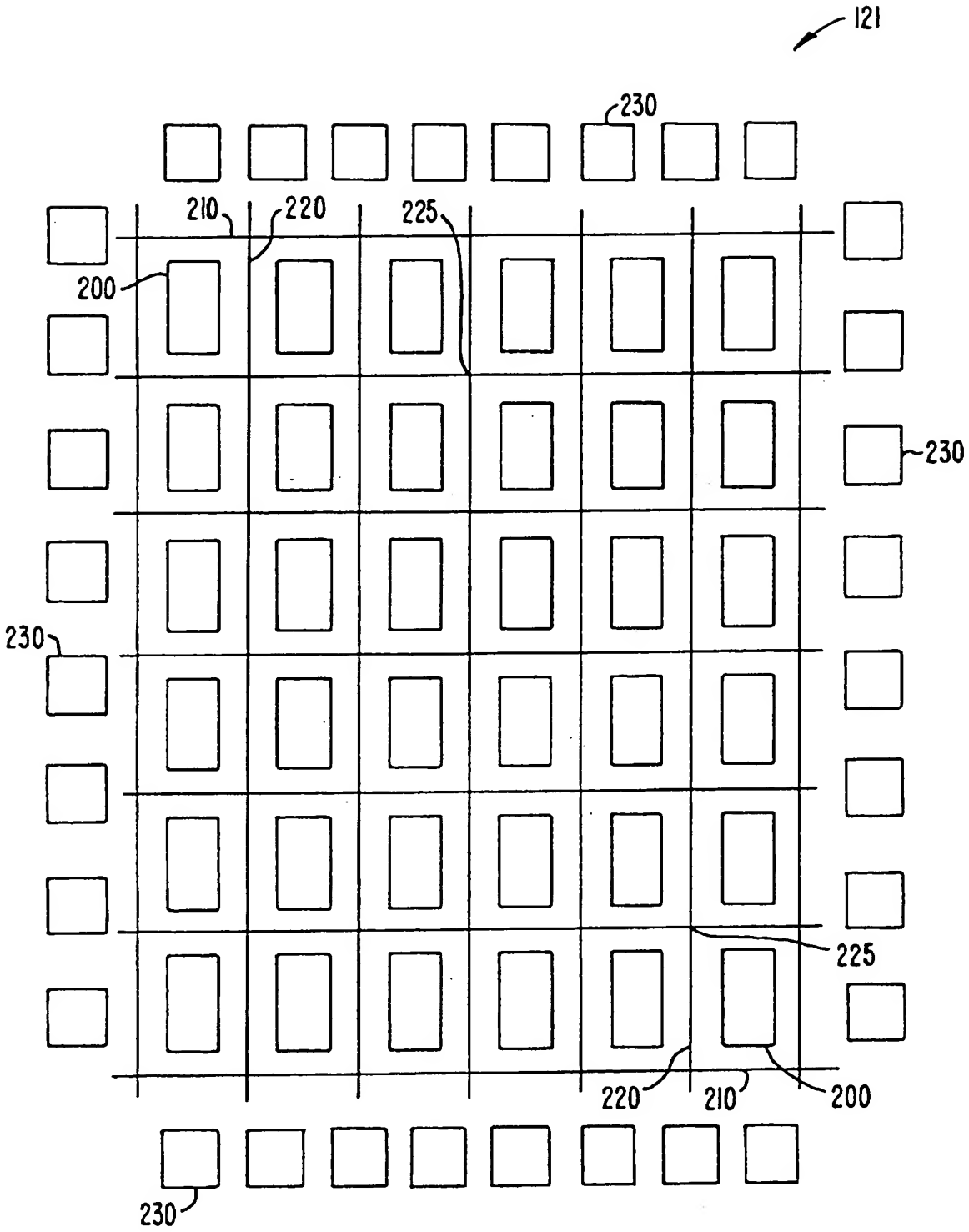

FIG. 12.

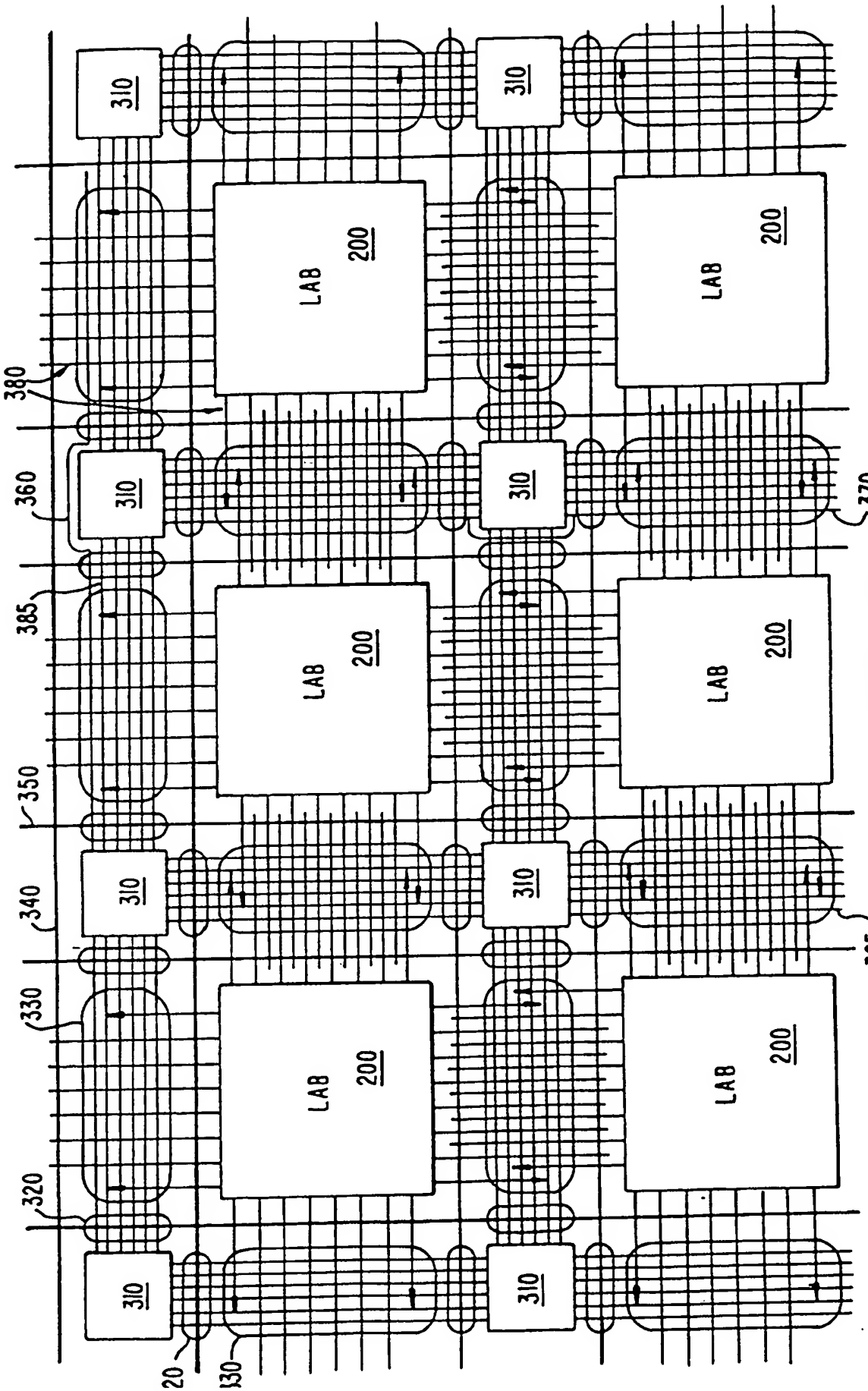

FIG. 13.

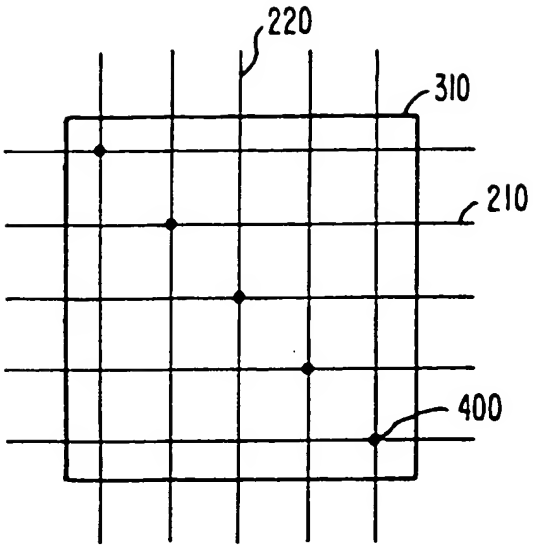

FIG. 14A.

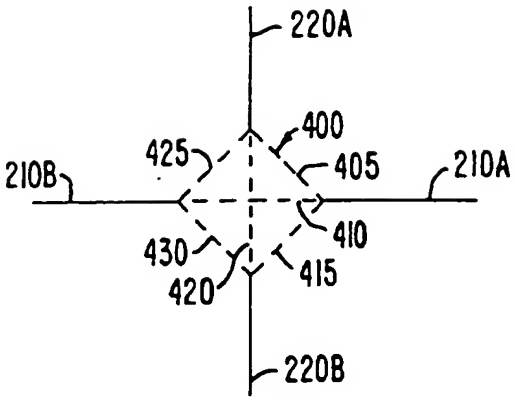

FIG. 14B.

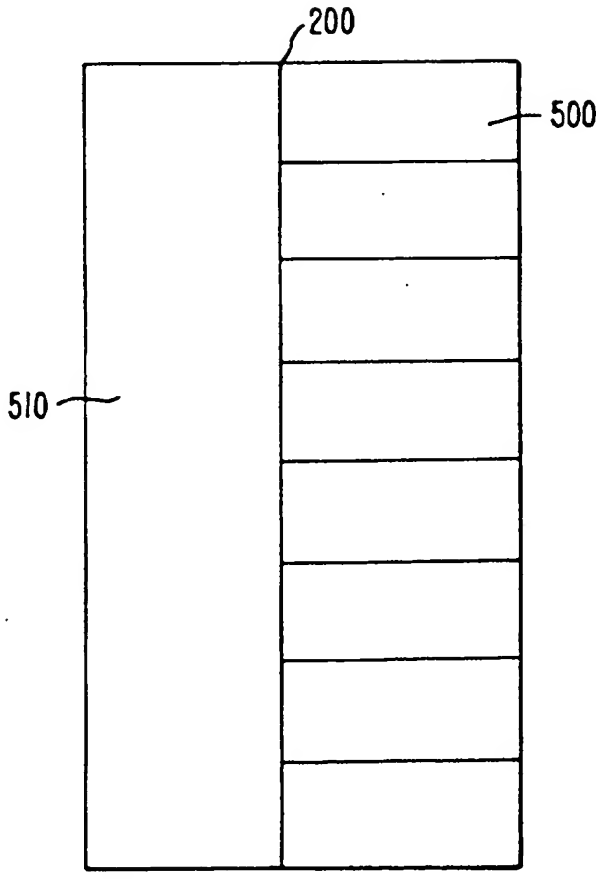

FIG. 15.

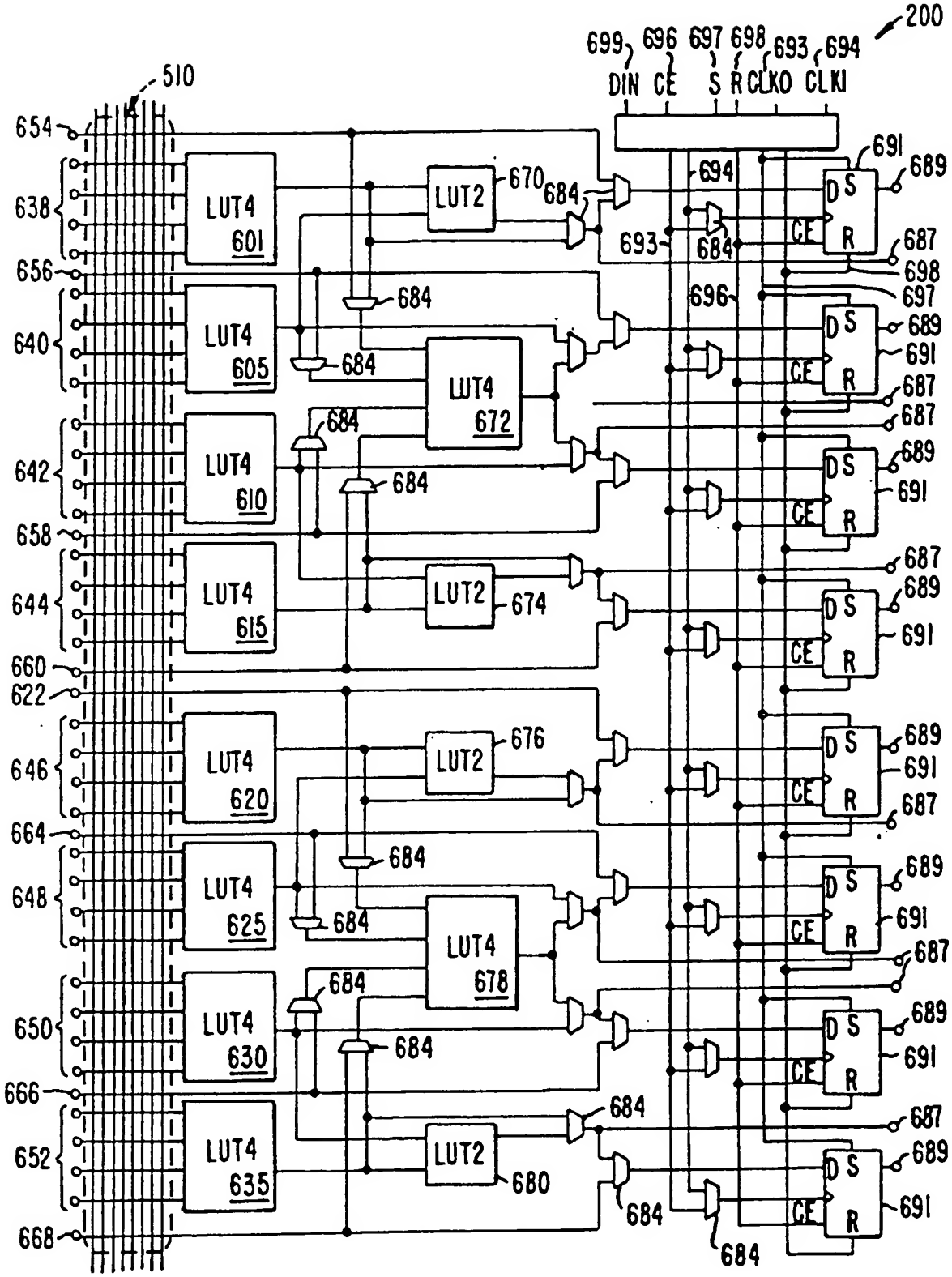
200
699 DIN
696 CE
697 S
698 R
693 CLKO
694 CLKI
510
654
638
656
640
642
658
644
660
622
646
664
648
650
666
652
668
LUT4 601
LUT4 605
LUT4 610
LUT4 615
LUT4 620
LUT4 625
LUT4 630
LUT4 635
LUT2 670
LUT4 672
LUT2 674
LUT2 676
LUT4 678
LUT2 680
684
691
689
687
698
697
696
693
694
D S
CE
R

FIG. 16.

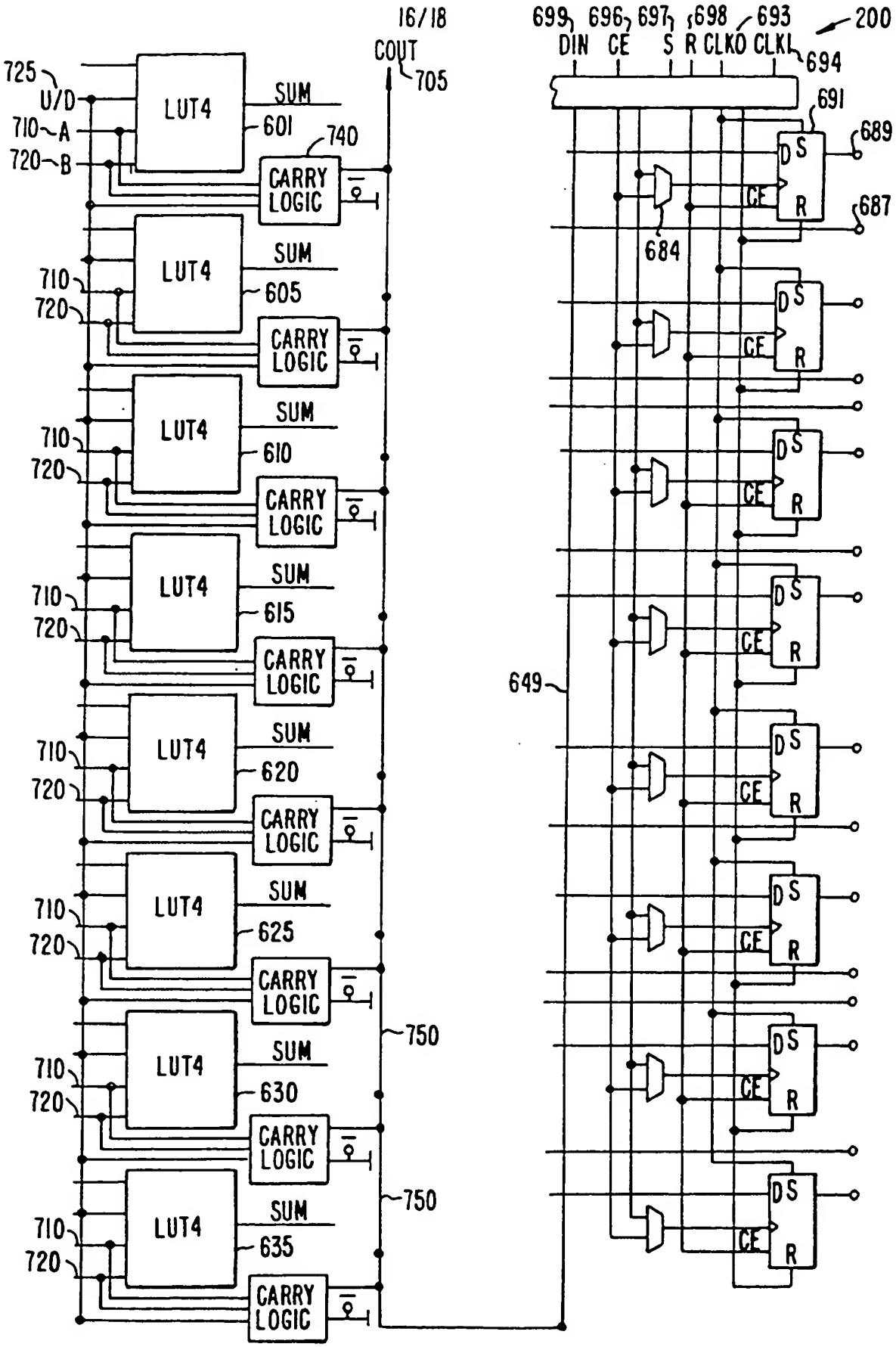
COUT
705
725
U/D
710
A
720
B
LUT4
SUM
601
740
CARRY LOGIC
605
610
615
620
625
630
635
750
649
699
DIN
696
CE
697
S
698
R
693
CLKO
CLKI
694
691
689
687
684
200
D
S
R

FIG. 17.

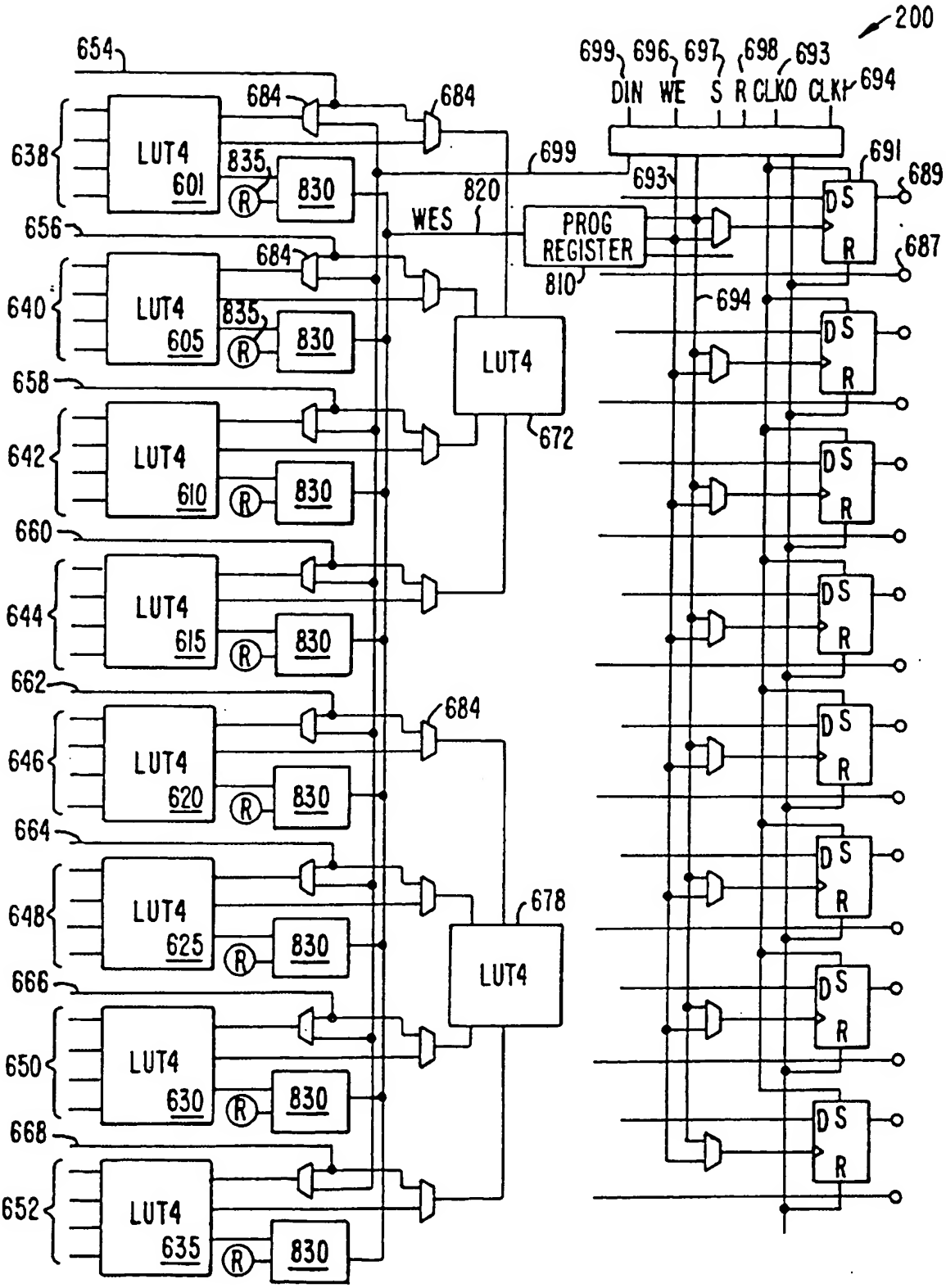
200
654
684
684
638
LUT4
601
835
R
830
699
699
696
697
698
693
694
DIN
WE
S
R
CLKO
CLKI
691
689
693
820
WES
PROG
REGISTER
810
687
656
684
694
640
LUT4
605
835
R
830
LUT4
658
672
642
LUT4
610
R
830
660
644
LUT4
615
R
830
662
684
646
LUT4
620
R
830
664
678
648
LUT4
625
R
830
LUT4
666
650
LUT4
630
R
830
668
652
LUT4
635
R
830
D
S
R

FIG. 18.

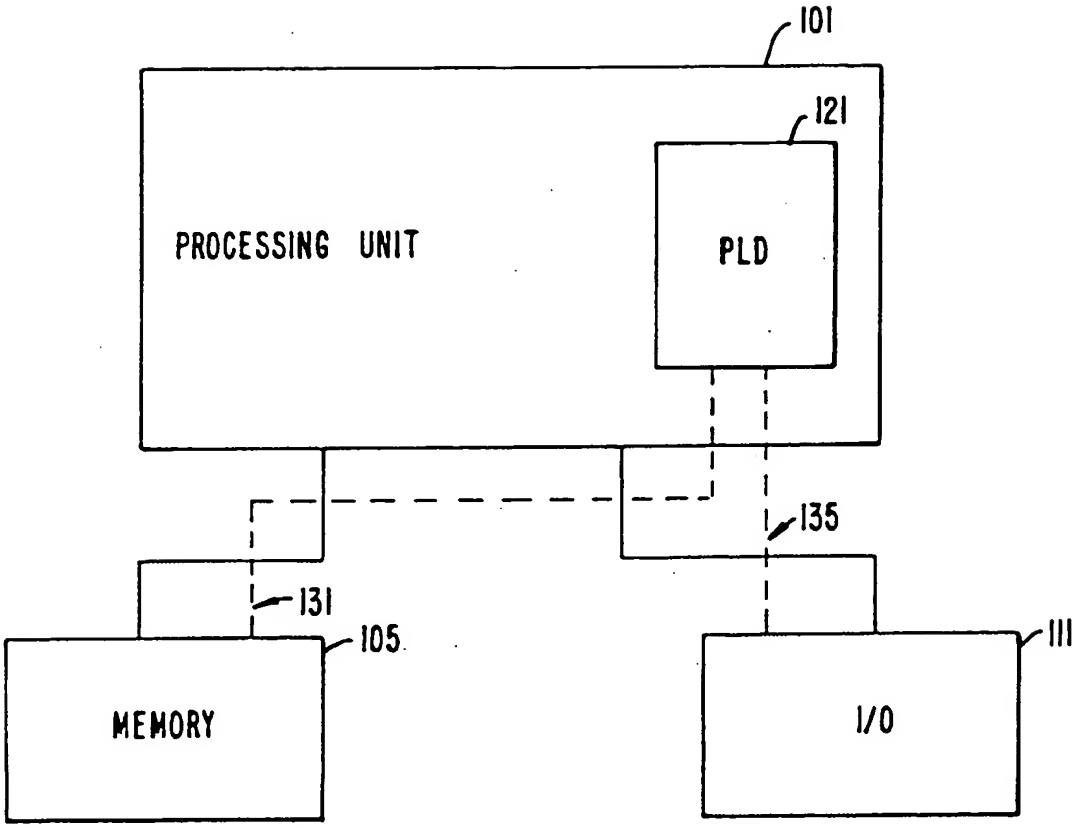

FIG. 19.

PROGRAMMABLE LOGIC ARRAY INTEGRATED CIRCUITS WITH ENHANCED ELEMENTS

This patent application claims priority from United States application Serial No. 08/334,879 filed November 4, 1994.

BACKGROUND OF THE INVENTION

The present invention relates to the field of digital electronic circuits. More specifically, the invention relates to an improved programmable logic device with enhanced elements as well as associated methods of operation.

Programmable logic devices (PLDs) are well known to those in the electronics art. Such programmable logic devices are commonly referred to as PALs (Programmable Array Logic), PLAs (Programmable Logic Arrays), FPLAs (Field Programmable Logic Arrays), EPLDs (Electronically Programmable Logic Devices), EEPLDs, LCAs (Logic Cell Arrays), FPGAs (Field Programmable Gate Arrays), and the like. Such devices are used in a wide array of applications where it is desirable to configure or program an off the shelf device for a specific application. Such devices include, for example, the well known Classic™ EPLDs, MAX® 5000 EPLDs, and FLEX® EPLDs all made by Altera®.

These devices, while similar in some aspects of overall functionality, may be of very different types in terms of circuit architecture. One family of PLDs uses a sum-of-products (SOP) architecture whereby each output is the ORed sum of a number of ANDed product terms of the inputs. This family is represented by the Altera MAX® and Classic™ 5000 EPLDs. Another family of PLDs uses look-up tables (LUTs) to perform logic functions. This family is represented by the Altera FLEX® EPLDs.

Modern PLDs generally are constructed from small functional units variously referred to as logic modules or macrocells and herein referred to as logic elements (LEs). These LEs are typically identical or nearly identical

throughout the PLD and perform a function that is a sub unit of the function of the entire PLD. For example, in a PLD based on an LUT architecture, the LEs might each be four input/one output LUTs. PLDs generally include an interconnect structure of conductors to provide a mechanism for selectively connecting the inputs and outputs of the LEs in order to perform the PLD functionality.

Larger PLD's of both the SOP and LUT type generally group the smaller LEs into larger functional units herein referred to as logic array blocks (LABs). The LABs can contain within them a local LAB interconnect that allows signals in one LE to be selectively connected to signals in a different LE in the same LAB and that transmits signals from the global interconnect to the inputs of the individual LEs. The LABs may be connected to one another and to input and output circuits by means of the global interconnect.

While such devices have met with substantial success, such devices also meet with certain limitations.

As one example of a limitation of the prior art, some prior art programmable logic devices (PLD) enhanced their functionality by providing "cascade" circuitry whereby the output of a logic element could be logically ANDed with the output of an adjacent logic cell. Such an arrangement allowed, for example, the output of a single logic element to be not only a function of its four inputs but to be any function of its four inputs ANDed with any function of the four inputs of its adjacent LE. While this circuitry allowed some additional functionality, its usefulness was limited. Because the cascade circuitry allowed combining of outputs only, only functions of more than four variables that have a function of up to four variables ANDed with another function of up to four variables can be implemented.

Another example of a limitation in prior art PLDs is that when PLDs are programmed to perform an addition function, they are not as efficient as could be hoped because carry operations use excessive resources. Specifically, according to one implementation of an ADD function in a PLD, a single LE is used for each bit of the ADD, and an additional LE is

needed for every third bit for carry generation. In addition, "expanders" are used for implementation of such ADD functions that provide "bit generate" and "bit propagate" signals, such as described in Langdon, Computer Design, 1982, pp. 494. Use of these logic resources for ADD operations not only reduces the capacity of the device to perform many functions, but also slows operation of the device. To allow for more efficient performance of ADD and related functions, PLD's have been equipped with a direct carry chain. The direct carry chain is an interconnect line that connects LEs in a LAB to adjacent LEs in that LAB and that directly connects a LAB to an adjacent LAB. An example of such a direct carry chain may be found in U.S. patent 5,260,611. The direct carry chain is illustrated in FIG. 2 of the '611 patent and labeled as elements direct carry_in line 70a and direct carry_out line 70b. While these direct carry chains have added enhanced functionality for PLD when computing ADD and related functions, having an exclusively direct carry chain has also added limitations in that functions utilizing the carry must be placed an adjacent LABs or on LABs that are directly connected to one another by the carry chain.

Another example of a limitation in prior art PLDs is that larger PLDs are generally provided with a large number of external input/output pins for transmitting signals off-chip, but the interconnect structure of these PLDs is such that a particular LE on the chip has direct access to a very limited number of I/O pins. A typical PLD may have as many as two hundred to four hundred or more I/O pins. Such devices are also provided with an intricate global interconnect structure for allowing individual functional elements on the PLD to communicate signals with all other elements on the PLD. For reasons of conserving chip area, this global interconnect structure is connected to the external pins in a limited way such that a particular LE on the chip has direct access to a very limited number of I/O pins. If a signal from a LE must be transmitted to an I/O pin to which that LE does not have direct access, the signal must be routed through an additional LE that has direct access to that I/O pin. This can create

problems in PLD reprogrammability and modifiability of certain logic designs. For example, if a minor modification in an application requires that a logic signal be routed to a different I/O pin, ideally the PLD should be easily reprogrammable to accommodate this modification. However, with the limitations of some prior art PLDs, rerouting a logic signal to a different I/O pin could require use of additional LEs. If these LEs were unavailable because they were being used in other parts of the design, modification of the chip to reroute the output signal might be impossible.

PLD's that are based on large numbers of identical LEs have certain limitations in programming flexibility based on the structure of the LE. A typical size for an LE is a four-input look-up table. This size look-up table is optimum for many functions. However, certain logic functions must use several LEs because of the limitations of the four-input LUT design. One modification to a four-input LUT that has been proposed is the partitioning of the LUT into smaller LUTs. These smaller LUTs can perform all of the logic function of the larger LUTs but also offer added flexibility. One such partitioning scheme was discussed in co-assigned U.S. Patent 5,274,581 which was co-invented by the inventor of the present invention. In that patent, it was discussed that a four-input LUT could be implemented as two three-input LUTs with various multiplexing circuitry. The LE discussed in that patent provided enhanced functionality when LEs where used as adders or counters. The discussed LE was not, however, optimized to perform multiplexing functions or more complex LAB-wide functions.

A further limitation of the prior art is that PLD's with a LAB-based structure typically group a number of LEs together into one LAB, in a specific embodiment each LAB has eight LEs. Each LE generally has one output which is a programmable function of its inputs. A typical number of inputs for an LE is four. A typical prior art LAB thus has eight outputs, one output for each of its eight LEs. In such a prior art LAB, the eight outputs are generally not combinable in that LAB. If a user wishes to have a function

that is a logical combination of the outputs of a LAB, the outputs of that LAB must be routed through the global interconnect to another LAB in the PLD and made the inputs to another LE which may then combine the signals. This results in an additional level of delay for these logic signals and in use of additional LEs on the PLDs which then can't be used for other functions.

Other limitations or prior art PLD's arise especially in situations in which the provision of additional or alternative types of interconnections between the logic modules would have benefits sufficient to justify the additional circuitry and programming complexity. Such additional interconnection paths may be desirable for making frequently needed kinds of interconnections, for speeding certain kinds of interconnections, for allowing short distance connections to be made without tying up a more general-purpose interconnection resource such as long-distance interconnect. There is also a continuing demand for logic devices with larger capacity. This produces a need to implement logic functions more efficiently and to make better use of the portion of the device which is devoted to interconnecting individual logic modules.

From the above it is seen that an improved programmable logic device is desired.

SUMMARY OF THE INVENTION

The invention provides an improved programmable logic device for use in digital circuits and systems having one or more enhancements.

According to one embodiment of the invention, the output of a logic cell may be logically ANDed with the input of an adjacent logic cell. The new cascade feature provided by the invention allows for PLDs with four-input LEs to use individual LEs to implement many functions of five variables.

In a further embodiment, the invention provides an improved programmable logic device in which a carry_in signal to a LAB may be provided by the carry_out of an adjacent LAB

or alternatively may be provided by a signal from the general interconnect. The LAB direct carry_out similarly may be connected to the general interconnect and may be provided to an adjacent LAB. The flexibility in carry chain routing provided by the invention allows for PLDs that are much improved in their ability to perform certain addition and counter functions. This improves the routing of carry chain designs by eliminating the requirement that a carry chain which spans multiple LABs be placed in adjacent LABs. It also improves the functionality of the device by eliminating the requirement of an extra logic element to get a carry_in to the first bit of a counter or get the carry_out from the last bit of a counter.

In a further embodiment, a LAB may be provided with lines in its LAB interconnect that are directly connected to an external I/O pin driver. This design allows the output pin routing to take advantage of the flexibility of routing of signals into the LABs that is built into the PLD. The flexibility in output routing provided by the invention allows for PLDs that are much improved in their ability to accommodate applications that require particular output pin assignments. This improves the routing of output signals by not requiring that a signal going to a particular I/O pin be directed through a particular group of LEs. It also improves the functionality of the device by not requiring an extra logic element to get an output signal to a particular I/O pin and allows direct connection between an input signal on one I/O pin and an output signal on another I/O pin without routing the signal through a LE.

In a further embodiment, the invention provides an improved programmable logic device for use in digital circuits and systems in which a LE is implemented as at least two LUTs and where LUTs are connectable to one or more LAB-wide input signals. This design allows for far larger and more complex logic functions to be implemented in a single LAB than in prior art PLDs.

In a further embodiment, the invention provides an improved programmable logic device for use in digital circuits

and systems in which a LAB is provided with one or more selectable wide-input AND gate on its outputs for selectively combining any of its output signals. This design allows for far larger and more complex logic functions to be implemented in a single LAB than in prior art PLDs. This improves the routing and programmability of LABs by not requiring that output signals from an LAB be routed to another LAB in order to be logically combined.

In accordance with the teaching of a further embodiment, a logic array block for a programmable logic device is provided including the following elements: a dedicated input; a first programmable function block for implementing logic functions; and a second programmable function block, selectively coupled to the dedicated input or the first function block, for implementing logic functions. More specifically, the logic array block for a programmable logic device includes: (1) a dedicated input; (2) a first four-input function block for implementing logic functions; (3) a second four-input function block for implementing logic functions; and (4) a first multiplexer block. This first multiplexer block has a first input, coupled to the dedicated input; a second input, coupled to an output of the first four-input function block; and an output, coupled to the second four-input function block. The first multiplexer block is for selectively coupling the dedicated input or the first four-input function block to the second four-input function block.

A computer system or other digital processing machine incorporating the invention will benefit from the added flexibility and reprogrammability of the PLD.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a simplified block diagram of a prior art PLD.

Fig. 2 is a simplified block diagram of a PLD showing an enhanced input cascade chain according to the invention.

Fig. 3 is an alternative simplified block diagram showing two LEs with an enhanced cascade and omitting many of the details of Fig. 2.

Fig. 4 is a simplified block diagram showing a LAB with enhanced carry routing and omitting many of the details of Fig. 2.

Fig. 5 is a schematic block diagram showing a LAB with LAB-based output routing according to the invention.

Fig. 6 is a simplified block diagram of a specific PLD showing the locations of I/O pins.

Fig. 7 is a simplified block diagram of a prior art PLD showing the routing of LE outputs to the I/O pins.

Fig. 8 is a simplified block diagram showing a LAB with LAB-based output routing according to the invention.

Fig. 9 is a schematic block diagram of part of a logic array block with a partitioned logic element and shared input signals according to the invention.

Fig. 10 is a schematic block diagram of a logic element with multiple look-up tables and shared input signals according to the invention.

Fig. 11 is a simplified block diagram showing a LAB-based wide-input AND gate according to the invention.

Fig. 12 is a block diagram showing the overall architecture of a programmable logic device;

Fig. 13 is a block diagram of an overall interconnection scheme for logic array blocks of the present invention;

Fig. 14 is a more detailed diagram of a switch box of the present invention;

Fig. 15 is a simplified block diagram of a logic array block (LAB) of a programmable logic device;

Fig. 16 is a more detailed block diagram of a logic array block of the present invention;

Fig. 17 is a block diagram showing a carry chain scheme of the present invention; and

Fig. 18 is a block diagram showing a logic array block of the present invention configured as a random access memory.

Fig. 19 is a block diagram of a digital processing system including a PLD with enhancements according to the invention.

DESCRIPTION OF THE INVENTION

For the sake of clarity, the invention will be explained with reference to a PLD architecture that was disclosed in co-assigned U.S. patent 5,260,611 (the '611 patent). However, it will be apparent to one skilled in the art that the invention may be used in a very wide variety of different PLD architectures including architectures that are fundamentally different from that disclosed in the '611 patent.

Fig. 1 shows the overall organization of an illustrative prior art PLD integrated circuit 10. Many details of this circuit not necessary for an understanding of the present invention are discussed in co-assigned U.S. patent 5,260,611. Not all of the conductors employed in circuit 10 are shown in Fig. 1, but enough is shown in this figure for the present discussion. Each LE 12 is represented by a small square in Fig. 1. LEs 12 are grouped together in groups of eight in this particular example. Each of these groups is referred to as a logic array block (LAB) 14. LABs 14 are arranged in six horizontal rows and twenty two vertical columns on circuit 10. Accordingly, there are a total of one hundred thirty two LABs 14 and one thousand fifty six LEs 12 on circuit 10. Each LE 12 is capable of performing a relatively elementary logic function, but extremely complex logic can be performed by variously interconnecting the LEs.

The interconnection circuitry shown in Fig. 1 includes groups of so-called global horizontal conductors (or global horizontal interconnect) 20 interspersed between the horizontal rows of LABs and groups of global vertical conductors (or global vertical interconnect) 22 interspersed between the vertical columns of LABS. These conductors are global in the sense that they extend along an entire row or column of LABs and are not localized to one LAB. Programmable interconnections can be made between select intersecting

horizontal and vertical conductors in order to apply signals on the vertical conductors to the horizontal conductors. Although other numbers of conductors can be used if desired, in the depicted embodiment, there are one hundred seventy six conductors in each group of global horizontal conductors 20 and sixteen conductors in each group of global vertical conductors 22. In one specific embodiment, programmable connections are possible between a vertical conductor 22 and just one horizontal conductor 20 in each of the six horizontal rows that the vertical conductor crosses. This limitation is imposed on the general interconnect because the circuitry needed to selectably allow a connection between intersecting conductors takes up a certain amount of area on the chip and this chip area is limited.

The interconnection circuitry shown in Fig. 1 further includes groups of vertical LAB input conductors 24 associated with each LAB 14 for conveying signals from the global horizontal conductors 20 intersected by conductors 24 into the LEs 12 in that LAB. The conductors 24 are part of the LAB interconnect which is local to just one LAB. Finally, the interconnection circuitry shown in Fig. 1 includes a set of universal fast interconnect conductors 30 which are globally connected to all LABs and are designed to transmit fast signals throughout the PLD. Although other numbers of conductors can be used if desired, in the depicted embodiment there are twenty four conductors in each group of LAB input conductors 24 and there are four universal fast conductors 30.

Turning now to Fig. 2, which shows one possible implementation of part of a typical LAB 14 incorporating enhancements according to the invention, three representative LEs 12 are shown with labelling LE1, LE2, LEN indicating that any number of LEs could be included in various embodiments of a LAB. Although LEs 12 can be implemented in other ways (e.g., as product-term-based macrocells), in the illustrative embodiment shown in Fig. 2, each LE 12 includes a look up table or universal logic block ("ULB") 40 and a flip-flop type device 142.

In addition to the above-described representative LEs 12, Fig. 2 shows portions of representative global horizontal conductors 20, global vertical conductors 22, local LAB input conductors 24, local inter-LAB feed-back conductors 26, LAB-input lines 28, and universal fast conductors 30. Each of LAB input conductors 24 can be connected to a selected one (or more) of conductors 20 and 30 via a programmably controlled programmable logic connector ("PLC") 50 (only some of which are shown in Fig. 2). Local inter-LAB conductors 26 connect only to inter-LAB signals and are used primarily for feed-back of LE outputs through lines 54 to the inputs of other LEs in the same LAB.

PLCs 50 can be implemented in any of a wide variety of ways. For example, each PLC 50 can be a relatively simple programmable connector such as a plurality of switches for connecting any one of several inputs to an output. Examples of components suitable for implementing PLCs 50 are EPROMs, EEPROMs, pass transistors, transmission gates, antifuses, laser fuses, metal optional links, etc. The components of PLCs 50 can be controlled by various function control elements ("FCEs") as described in more detail below (although with certain PLC implementations (e.g., fuses and metal optional links) separate FCE devices are not required, so that in those cases the depiction of FCE devices in the accompanying drawings merely indicates that the PLCs are programmable). In the depicted embodiment each PLC 50 is a 15-to-1 multiplexer which is controlled by programmable function control elements ("FCEs") 51 to connect one of its fifteen inputs to its output. Each of the fifteen inputs is the signal on a predetermined respective one of conductors 20 or 30. There is one PLC 50 for each of the twenty four LAB input conductors 24. Each of conductors 20 and 30 is connected to two of PLCs 50. Accordingly, each of conductors 20 and 30 is connectable to two of conductors 24. The letter P and the associated arrow symbol inside dotted line 49 indicate that the population of connections from conductors 20 and 30 to the inputs of each of PLCs 50 is a partial population meaning that not every intersection indicates a possible interconnection.

Like PLCs 50, FCEs 51 can also be implemented in any of several different ways. For example, FCEs 51 can be SRAMs, DRAMs, first-in first-out ("FIFO") memories, EPROMs, EEPROMS, function control registers ferro-electric memories, fuses, antifuses, or the like.

Finally, Fig. 2 shows direct carry_in line 70a, direct carry_out line 70b, LAB-input carry_in line 71a, global interconnect carry_out line 71b, carry_in selector or PLC 80 with its associated FCE 82. The operation of these elements will be described further below.

Each of the four data inputs to each LE 12 can be connected to any one (or more) of LAB input conductors 24 and/or any one (or more) of local conductors 26 via a PLC 52. PLCs 52 (and similar PLCs 34 which are discussed below) may have any of the characteristics described above for the general case of PLCs 50. However, in the depicted, presently preferred embodiment each of PLCs 52 (and each similar PLC 34) is a 32-to-1 mux so that any one of the twenty-four LAB input conductors 24 or any one of the eight local conductors 26 can be connected to each LAB data input. Accordingly, the letter F and the associated arrow symbol inside chain dotted line 59 indicate that the population of connections from conductors 24 and 26 to PLCs 34 and 52 is a full population meaning that each intersection of lines indicates a possible interconnection. With these interconnections just described, any of the signals on the global horizontal lines 20 in that row or on universal fast interconnects 30 may be connected to any of the LE inputs. To avoid overcrowding the drawing, the FCEs (similar to above-described FCEs 51) for programmably controlling PLCs 34 and 52 are not shown separately, but rather are assumed to be part of elements 34 and 52.

The data output of each LE 12 can be applied (via conductor 54) to a respective one of local conductors 26. These conductors serve only the LEs in the associated LAB 14. In other words, conductors 26 do not extend beyond the associated LAB. The data output of LE 12 can also be applied to either or both of two global vertical conductors 22 (via tri-state drivers 56), and to one of global horizontal

conductors 20 (via PLC 58 and tri-state driver 60). The other input or inputs to each PLC 58 are the signals from one or more of global vertical conductors 22. Accordingly, global vertical conductors 22 are selectively connectable to global horizontal conductors 20 via PLCs 58 and tri-state drivers 60. PLCs 58 may be similar to any of the above-described PLCs. The use of tri-state drivers 56 and 60 is advantageous to strengthen signals which may be required to travel relatively long distances throughout circuit 10 and/or which may be inputs to relatively large numbers of other components. Some or all of tri-state drivers 56 and 60 may be controlled (i.e., turned on or off) by FCEs (e.g., FCEs 57) on circuit 10. PLCs 58 are also typically controlled by FCES on the circuit. All of these FCEs may be similar to above-described FCEs 51.

In addition to being available as data inputs to LEs 12, the signals on any of conductors 24 and 26 can also or alternatively be applied to any of local vertical conductors 32 via PLCs 34. In the depicted embodiment, each of PLCs 34 is a 32-to-1 switch, but PLCs 34 can alternatively have any of the characteristics described above for the general case of PLCs 50. Although only two PLCs 34 are shown in Fig. 2, there is preferably one such PLC for each of the four conductors 32. Each of conductors 32 is connectable to any one of universal fast conductors 30 for receiving the signal on the fast conductor. These connections from conductors 30 to conductors 32 are preferably made in the same way that the connections from conductors 24 and 26 to conductors 32 are made, i.e., by PLCs controlled by FCEs (all represented by element 31 in Fig. 2). Again, although each of these PLCs can have any of the characteristics described above for the general case of PLCs 50, in the depicted, presently preferred embodiment each of these PLCs can connect any of conductors 30 to an associated one of conductors 32. The letter F and the arrow symbol inside chain dotted line 31 indicate that the population of possible connections from conductors 30 to each of conductors 32 is a full population. Each of conductors 32 is connectable (via conductors 36) to each of LEs 12. FCE-controlled PLCs in each LE allow these signals to be used for such purposes as

flip-flop clock and flip-flop clear. Thus local vertical conductors 32 are known as clock and clear lines and can be driven from fast lines 30 for synchronous clocks and clears (i.e., clocks and clears which come from outside of device 10 and are available everywhere throughout device 10), or from LAB input lines 24 or local lines 26.

According to the invention, a LAB in PLD 10 is provided with cascade connect conductors associated logic gates 94 and selectors or PLCs 90. These elements operate together to give the additional functionality of the invention described in more detail in the discussion of Fig. 3. They allow the data output of a LE to be logically combined with the input of an adjacent or nearby LE 12, if desired, to perform more complex logic functions without the need to pass intermediate data through the general interconnection network. The elements are an enhancement over the cascade conductors disclosed in commonly assigned, co-pending application Serial No. 07/880,888 incorporated herein by reference for all purposes.

Fig. 3 illustrates two LE's 12 according to the invention. While the discussion so far has suggested use of these LEs in a LAB, the invention may be employed in PLD devices that do not group LEs into LABs. A cascade chain is represented by a cascade connector 70a. According to the invention, the cascade chain allows the output of a LE to be logically combined with one of the input lines to the LE before that input signal is input to the LE. In a specific embodiment, this is accomplished through use of cascade selector 90. Cascade selector 90 is controlled by a FCE 92 such that one of the signals to logic gate 94 may be either the output of an adjacent LE or may be a constant (designed to not effect the output of logic gate 90). In a specific embodiment, logic gate 94 performs AND function and one of the inputs of selector 90 is permanently connected to a true logic level.

In an embodiment where the LEs are grouped in LABs, each LAB may be provided with a cascade-out connector 72c to enable the output of the last LE in that LAB to be cascaded to

a LE in an adjacent or nearby LAB through a cascade-in connector 72b. In a further embodiment, cascade-out connector 72c may be connectable to the general interconnect and cascade-in 72b may be connectable to the general interconnect or to the LAB interconnect.

As discussed above, prior art PLDs with output cascade circuitry allowed, for example, the output of a single LE to be not only a function of its four inputs but to be any function of its four inputs ANDed with any function of the four inputs of its adjacent LE.

As illustrated in Fig. 3, the present invention provides a means further enhancing a PLD by ANDing one of the inputs to a LE with the output of an adjacent LE. Since the adjacent LE now feeds one of the inputs to the look-up table, more complex functions are possible than just a simple "AND." The invention can thus be used to implement many functions of five variables. By tying input "A" in Fig. 3 to VCC, the invention can also implement all four input functions where one of the inputs is a "fast" cascade input from the adjacent LE.

Returning to Fig. 2, Fig. 2 further illustrates an enhanced carry chain according to the invention. The direct carry chain is represented in part by conductors 70a and 70b. These interconnections allow a carry_out output of each LE 12 to be used as a carry_in input to an adjacent or nearby LE as shown, for example, in commonly assigned, co-pending patent application Serial No. 07/880,752 incorporated herein by reference for all purposes. For example, carry chain conductors 70a allow the carry_out output of each LE 12 shown in Fig. 2 to be the carry_in input to the next higher LE in that Fig. Similarly, carry chain conductor 70b runs from the top-most LE 12 in the LAB fragment shown in Fig. 2 to the bottom-most LE in the horizontally adjacent LAB in the adjacent column of LABs. This allows the carry chain to continue from LAB to LAB if desired. According to the invention, the carry routing is enhanced through use of the support circuitry shown in more detail in Fig. 3.

FIG. 4 illustrates one LAB 14 according to the invention. LAB 14 receives input signals from interconnect 24 and generates output signals to horizontal interconnect 22. Each LAB 14 includes N programmable logic elements 12, and each LE has associated with it carry logic 13. A direct carry chain is represented by a direct carry_in 70a and a direct carry_out 70b. According to the invention, the carry_in may also be connected to the global interconnect. In a specific embodiment, this is accomplished through use of carry_in selector 80. Carry_in selector 80 is controlled by a FCE 82 such that the carry_in to LE 1 may come from either direct carry_out 70b from an adjacent LAB or may come from one of the signals in the global interconnect through LAB-input carry_in line 71a, depending on the state of FCE 82. The line 71a that optionally connects to the carry_in of a LAB may be dedicated to this purpose or may be shared by other signals such as a LAB-wide clock signal as illustrated by the additional connection 75 in Fig. 2.

At the carry_out of a LAB according to the invention, the carry_out from the last LE in the LAB is provided to the direct carry chain through direct carry_out line 70b and is also provided to the global interconnect through global interconnect carry_out line 71b.

Fig. 5 shows one possible implementation of part of a typical LAB 14 as also illustrated and described with reference to Fig. 2 but incorporating enhancements to the output routing according to one embodiment of the invention wherein one or more additional lines 55 are included in the area 59 of full population interconnections in one or more of the LABs of the device. As with the LE inputs described in reference to Fig. 2, any of the signals on the global horizontal lines 20 in that row or on universal fast interconnects 30 may be connected to any of these additional lines 55. However, instead of these lines 55 being connected to an LE input or other signal in the LAB, each is connected to an external device I/O pin 100, optionally, the lines 55 may be connected to the output pins through a driver that is not shown. The importance of this additional interconnection

will be more fully understood with reference to the additional figures discussed below.

Fig. 6 shows how input/output pins may be provided on circuit 10. Six groups 90 of eight input/output pins are shown along each side of circuit 10. Twenty-two groups 92 of two input/output pins are shown along each of the top and bottom of the circuit. In addition, there are four fast input pins 94 respectively connected to fast conductors 30, seven control pins 96 for use during programming of device 10, and approximately thirty supply pins 98 for supplying power and ground potential to device 10.

Fig. 7 shows an illustrative embodiment of a prior art arrangement for connecting LE output signals to I/O pins. In Fig. 7 the one hundred seventy six global horizontal conductors 20 associated with each horizontal row of LABs 14 are arranged so that half can receive the outputs of the drivers 60 (Fig. 2) associated with the top four LEs 12 in each LAB in that row, while the other half of these conductors 20 can receive the outputs of the drivers 60 associated with the lower four LEs 12 in each LAB in that row. Accordingly, in Fig. 7 each group of one hundred seventy six global horizontal conductors 20 is shown divided into two subgroups of eighty-eight such conductors, one subgroup being above the associated row of LABs 14, and the other subgroup being below that row of LABs. Each input/output pin 100 in each group 90 of such pins can receive an output via a PLC 102 from a subset of the conductors 20 in a subgroups of eighty-eight such conductors. PLCs 102 can have any of the characteristics described above for the general case of PLCs 50, and PLCs 102 are controlled by programmable FCEs similar to those described above. One of these two subgroups of conductors is associated with the lower LEs 12 in one row of LABs 14, while the other subgroup of conductors is associated with the upper LEs 12 in the LAB row immediately below the first row.

For use as an input pin, each pin 100 is connected through two tri-state drivers to two of conductors 20. Thus each symbol 104 in Fig. 7 represents two tri-state drivers,

each of which is programmably controlled (e.g., by an FCE similar to those described above).

One important aspect this prior art circuit for an understanding of the present invention is that all connections to input/output pins 100 are made through the global interconnect via lines 20. For output in this prior art circuit, each global interconnect line 20 can connect to only two particular I/O pins and each vertical interconnect 22 can connect to only two I/O pins. As explained in the discussion of Fig. 2, an output of a LE 12 can connect to only one horizontal interconnect 20 and to two vertical interconnects 22. The LE can therefore directly connect to only thirty output pins: two on its own horizontal line, four on its two vertical lines, and 12 on the six horizontal lines that are accessible from the two vertical lines. If the output signal from an LE is required at a different pin from one of those thirty, the signal must be routed through a different LE.

The LAB illustrated in Fig. 5 according to the invention avoids this limitation by allowing an output pin to be selectively connected to any of the LAB-input lines 24 or inter-LAB lines 26 via PLCs 55. As described above, at least one of LAB-input lines 24 may be connected to any one of the one hundred seventy-six horizontal global conductors 20 associated with that LAB's row. With this interconnection, the output of any LE on the PLD may be connected to any output pin having a LAB-based connection without the need to pass through any other LE.

Fig. 8 is a simplified block diagram of a LAB 14 showing only those details necessary to illustrate the invention. Fig. 8 illustrates a LAB 14 having eight LEs 12 and having LAB-input lines 24. Two LAB-based output lines are shown connected to PLCs 55 which drive the signals to output pins 100.

Fig. 9 shows one possible implementation of part of a typical LAB 14 as also illustrated and described with reference to Fig. 2 but incorporating enhancements to the output routing according to one embodiment of the invention wherein LEs 12 each include two LUTs 40a and 40b as well as

associated circuitry that is shown in more detail in Fig. 10. Fig. 9 also illustrates LAB-wide input signals 43a and 43b which are input signals that are shared among the LEs 12 in LAB 14. These shared signals are connectable to the local interconnect through programmable logic connectors 34.

Fig. 10 is a simplified block diagram of a LE 13 showing only those details necessary to illustrate the invention. In this aspect of the invention, the look-up table of a logic element has been partitioned into two 3-input look-up tables 40a and 40b whose inputs can either be connected to one of the standard input lines dedicated to the logic element, or, according to the invention, be connected to one of the global LAB-lines 43a or 43b that feed multiple LEs 12.

According to an embodiment of the invention illustrated in Fig. 10, LE 12 is equipped to receive inputs A, B, C, and D that are assigned to that one LE and are unshared with other LEs. LE 12 is also equipped to receive shared inputs X and Y that are shared among several to all LEs in a LAB. The six inputs are selectively connectable to the inputs of look-up tables 40a and 40b via three input multiplexers 44a-c. Input multiplexer 44a selects between inputs A and Y and is connected to look-up table 40a. Input multiplexer 44b selects between inputs B and D and is connected to look-up table 40a. Input multiplexer 44c selects between inputs C and Y and is connected to look-up table 40b. A fourth input multiplexer 44d selects between inputs D and X and is connected to the select signal of an output multiplexer 46. Output multiplexer 46 selects between the outputs of look-up tables 40a and 40b and connects one of those outputs to the output of LE 12. In the depicted embodiment, the input multiplexers have their select lines connected to a programmable store or FCE element. FCE 45a provides the select signal for multiplexers 44a and 44c. FCE 45b provides the select signal for multiplexers 44b and 44d.

According to an embodiment of the invention as illustrated in Figs. 9 and 10, a logic element 12 can be configured to be a function F(a,b,c,d), as would be standard for four-input LUTs in the prior art, or as F(y,b,c)&d #

G(a,b,y)&!d, or as F(a,b,c)&x # G(a,d,c)&!x, or as F(a,b,y)&x # G(c,d,y)&!x, where a,b,c and d are signals from the logic element's dedicated input lines, while x and y are signals provided according to the invention that are global to the LAB. The combination according to the invention of a LE constructed of partitioned LUTs and of a number of LAB-wide input signals allows functions of greater complexity to be implemented in a single LE as compared to a standard prior art LE. For example, a four-input multiplexer selectable by signals X and Y could fit on a single logic element built according to this aspect of the invention, whereas two prior art LEs would be required.

Returning now to Fig. 9, according to the invention, at least one LAB 14 is provided with a wide input AND gate 74. This AND gate is connectable to the outputs of up to all of the LEs 12 in LAB 14 as shown. The LE 12 output lines are connectable to AND gate 74 via XOR gates 76 which provide a selectable inversion function under control of FCE 76a and OR gates 78, which allows selectable connection of any of the LE outputs to AND gate 74. The details of these improvements according to the invention are explained further with reference to Fig. 11. In one embodiment of the invention, the output of AND gate 74 is connectable to lines in the global interconnect 20 and 30 as shown. In a further embodiment, AND gate 74 may include one or more input lines 79 connectable to the LAB-based interconnects 24 or 26. This embodiment would allow the outputs of the LEs to be further logically combined with a signal from the global interconnect via the LAB interconnect.

Fig. 11 is a simplified block diagram of a LAB 14 showing only those details necessary to illustrate the wide-input AND gate according to the invention. Fig. 11 illustrates a LAB 14 having eight LEs 12 and having LAB-input lines 24. A LAB-based wide AND output gate 74 is shown connecting the eight LE output lines. Each of the output lines is connectable to AND gate 74 via an XOR gate 76 which provides a selectable inversion function under control of FCE 76a and an OR gate 78 with one of its inputs fed by FCE 78a,

which allows selectable connection of any of the LE outputs to AND gate 74, i.e., when the FCE 78a is in the true state, the output of OR gate 78 is always true regardless of the output of its associated LE 12. Therefore, when an FCE 78a is in the true state, the output of AND 74 gate does not depend on the output of the LE 12 associated with that FCE and that LE is effectively disconnected for the input of AND gate 74.

Fig. 12 illustrates an alternative embodiment and configuration for a PLD. Fig. 12 shows a six-by-six two-dimensional array of thirty-six logic array blocks (LABs) 200. LAB 200 is a physically grouped set of logical resources that is configured or programmed to perform logical functions. The internal architecture of a LAB will be described in more detail below in connection with Fig. 13. PLDs may contain an arbitrary number of LABs, more or less than the PLD 121 shown in Fig. 12. Generally, in the future, as technology advances and improves, programmable logic devices with even greater numbers of logic array blocks will undoubtedly be created. Furthermore, LABs 200 need not be organized in a square matrix; for example, the array may be organized in a five-by-seven or a twenty-by-seventy matrix of LABs.

LAB 200 has inputs and outputs (not shown) which may be programmably connected to a global interconnect structure, comprising an array of global horizontal interconnects (GHs) 210 and global vertical interconnects (GVs) 220. Although shown as single lines in Fig. 12, each GH 210 and GV 220 line represents a plurality of signal conductors. The inputs and outputs of LAB 200 are programmably connectable to an adjacent GH 210 and an adjacent GV 220. Utilizing GH 210 and GV 220 interconnects, multiple LABs 200 may be connected and combined to implement larger, more complex logic functions than can be realized using a single LAB 200.

In one embodiment, GH 210 and GV 220 conductors are programmably connectable at intersections 225 of these conductors. Moreover, GH 210 and GV 220 conductors may make multiple connections to other GH 210 and GV 220 conductors. Various GH 210 and GV 220 conductors may be programmably connected together to create a signal path from a LAB 200 at

one location on PLD 121 to another LAB 200 at another location on PLD 121. Furthermore, an output signal from one LAB 200 can be directed into the inputs of one or more LABs 200. Also, using the global interconnect, signals from a LAB 200 can be fed back into the same LAB 200. In other embodiments or the present invention, only selected GH 210 conductors are programmably connectable to a selection of GV 220 conductors. Furthermore, in still further embodiments, GH 210 and GV 220 conductors may be specifically used for passing signal in a specific direction, such as input or output, but not both. For example, one or more GH 210 or GV 220 conductors may be used as a dedicated input driver or dedicated clock network to drive the LABs 200 from an input pin of the integrated circuit.

The PLD architecture in Fig. 12 further shows at the peripheries of the chip, input-output drivers 230. Input-output drivers 230 are for interfacing the PLD to external, off-chip circuitry. Fig. 12 shows thirty-two input-output drivers 230; however, a PLD may contain any number of input-output drivers, more or less than the number depicted. Each input-output driver 230 is configurable for use as an input driver, output driver, or bidirectional driver. An input driver takes signals from outside the chip and interfaces them to on-chip circuitry. An output drive takes internal signals and interfaces them to the outside world. A bidirectional driver performs the functions of both a input driver and an output driver. In addition, a bidirectional drive has a high-impedance mode which allows the driver to interface with a bidirectional bus. In other embodiments of the present invention, a PLD may have dedicated input drivers and dedicated output driver, as well as special "fast" input drivers and the like.

Like LABs 200, input-output drivers 230 are programmably connectable to adjacent GH 210 and GV 220 conductors. Using GH 210 and GV 220 conductors, input-output drivers 230 are programmably connectable to any LAB 200. Input-output drivers 230 facilitate the transfer of data between LABs 200 and external, off-chip circuitry. For

example, off-chip logic signals from other chips may be coupled through input-output drivers 230 to drive one or more LABs 200. Based on these off-chip inputs and the logical functions programmed into LABs 200, LABs 200 will generate output signals that are coupled through the global interconnect to input-output drivers 230 for interfacing with off-chip circuitry.

Fig. 13 shows a further embodiment of an overall internal architecture and organization of PLD 121. PLD 121 of Fig. 13 includes LABs 200. Fig. 13 shows six LABs, however, PLD 121 may have any arbitrary number of LABs, more or less than shown in Fig. 13. Furthermore, PLD 121 may be organized in any arbitrary format such as a ten-by-twelve. The internal architecture of a LAB 200 will be described in more detail below.

LABs 200 of Fig. 13 are programmably connectable, as described above in Fig. 12, using global interconnection resources. As in Fig. 12, the global interconnection resources of Fig. 13 are also organized in horizontal and vertical directions. Using these global interconnection resources, LABs 200 may be programmably combined to form larger, more complex logic functions than available from a single LAB. The global interconnection resources of Fig. 13 specifically include switch boxes 310, partially populated multiplexer regions 320, half-populated multiplexer regions 330, horizontal long lines 340, vertical long lines 350, horizontal double lines 360, and vertical double lines 370.

Furthermore, Fig. 13 shows only a portion of PLD 121. PLD 121 may also contain input-output drivers 230 (not shown), as in Fig. 12, for interfacing with PLD 121 with off-chip circuitry. As in Fig. 12, input-output drivers 230 (not shown) and are programmably connectable using the global interconnection resources.

There are various types of interconnection resources, distinguishable on the basis of the relative length of their segments. In particular, long lines (also known as "global lines"), including horizontal long lines 340 and vertical long lines 350, are conductors which run the entire

length and width of the array. Horizontal long lines 340 extend in a first direction of an array of LABs 200. Vertical long lines 350 extend in a second direction of the array of LABs 200.

Horizontal and vertical long lines 340 and 350 are used to programmably couple signals across the entire PLD 121. In this fashion, multiple LABs 200 may be combined to implement larger, more complex logic functions. Furthermore, long lines 340 and 350 are suitable conductors for distributing high fan-out, time-critical control signals such as a clock signal throughout a PLD integrated circuit with minimal timing skew. Moreover, long lines 340 and 350 may be fashioned into a bidirectional, tristatable bus. In one embodiment, PLD 121 may include long lines dedicated for a particular function, such as a dedicated clock line for routing a clock network.

As shown in Fig. 13, LABs 200 have input-output lines 380 for receiving and providing logic signals. LAB inputs-outputs lines 380 include bidirectional paths, which may be programmed or configured as an input or an output. Furthermore, LAB input-output lines 380 may include dedicated inputs and dedicated outputs. Moreover, LAB input-output lines 380 may includes a combination of bidirectional paths, dedicated inputs, and dedicated outputs.

Using LAB input-output lines 380, horizontal and vertical long lines 340 and 350 may be used to programmably couple signals to and from LABs 200 in different locations of PLD 121. Specifically, long lines 340 and 350 can provide input signals for LAB 200 from other LABs 200. Long lines may also be driven by circuitry such as input-output drivers 230 (not shown). Input-output drivers 230 may be used programmably couple, through long lines 340 and 350, input signals from external, off-chip circuitry and sources to LABs 200.

Specifically, in one embodiment, dedicated outputs from LAB 200, via LAB input-output lines 380, may be programmably coupled directly, without passing through another global interconnection resource, to horizontal long lines 340.

In addition, LAB input-output lines 380 may also be programmably coupled indirectly to horizontal and vertical long lines 340 and 350 through other global interconnection resources including double lines 360 and 370.

To connect to the dedicated inputs of LAB 200, long lines 340 and 350 may be programmably coupled through partially populated multiplexer region 320 (at intersections of long lines 340 and 350 and double lines 360 and 370) to double lines 360 and 370. From double lines 360 and 370, signals may be programmably coupled through half-populated multiplexer region 330, to LAB input-output lines 380 of LAB 200. In other embodiments of the present invention, horizontal and vertical long lines 340 and 350 may be programmably coupled directly to the dedicated inputs of LAB 200 or selected LABs 200.

By not providing a direct programmable input path from long lines 340 and 350 to LABs 200, this reduces the amount of circuitry required in PLD 121. Overall die size of PLD 121 will be reduced without adversely affecting greatly the performance of the integrated circuit. The negative impact on performance will be minimal. For example, timing skew differences between different LABs 200 will be similar because the same delay will be introduced for the input signals into LAB 200. Furthermore, there will be some increases because less circuitry at the inputs of the LABs 200 also results in reduced parasitics such as resistances and capacitances, which tend to degrade performance.

In addition to horizontal and vertical long lines 340 and 350, PLD 121 of Fig. 13 includes double lines 360 and 370 for routing signals within PLD 121. Like long lines 340 and 350, double lines 360 and 370 extend in the horizontal and vertical directions of the array. Horizontal double lines 360 extend in the first direction of the array of LABs 200. Vertical double lines 370 extend in the second direction of the array of LABs 200. Compared to long lines 340 and 350, double lines 360 and 370 support shorter, local connections between two adjacent LABs 200 without using other global interconnection resources such as switch boxes

310 and long lines 340 and 350. To simplify the diagram in Fig. 13, only the referenced double lines 360 and 370 are shown bypassing switch box 310. Although not shown, other double lines in Fig. 13 also programmably couple two adjacent LABs 200 without using switch boxes 310.

As is the case with long lines 340 and 350, double lines 360 and 370 may be used to combine multiple LABs 200 to implement larger, more complex logic functions. Horizontal and vertical double lines 360 and 370 are used, for example, to programmably couple, through half-populated multiplexer region 330, input and output signals (via LAB input-output lines 380) of one LAB 200 to another LAB 200. This path does not pass through switch boxes 310, horizontal long lines 340, or vertical long lines 350. Since double lines 360 and 370 provide shorter-length interconnections than long lines 340 and 350, double lines 360 and 370 generally have better performance characteristics than long lines 340 and 350. Since long lines 340 and 350 are limited resources, using double lines 360 and 370 reserves long lines 340 and 350 for logic functions requiring longer-length signal paths.

Double lines 360 and 370 can drive or be driven by a LAB 200 which has LAB input-output lines 380 crossing, or intersecting, those particular double lines. More specifically, LAB input-output lines 380 may be programmably coupled to double lines 360 and 370 through half-populated input multiplexer region 330 at intersections of double lines and LAB input-output lines. As discussed above, long lines 340 and 350 may be programmably connected to double lines 360 and 370 through partially populated multiplexer regions 320 at intersections of long lines and double lines.

Double lines 360 and 370 may be programmably coupled to other double lines 360 and 370 via switch boxes 310, discussed below. In particular, to couple signals between more than two LABs 200, horizontal and vertical double lines 360 and 370 may be programmably coupled to another via switch boxes 310, as needed, to implement a particular logic function.

The present invention may include single lines 385, which are similar to double lines 360 and 370 except that these only intersect LAB input-output lines 380 of one LAB 200, instead of two. For examples, single lines 385 may be programmable coupled to other single lines 385 via switch boxes 310. Single lines 385 may drive or be driven by a LAB 200 which has LAB input-output lines 380 crossing, or intersecting, those particular single lines 385. In some embodiments of the present invention, however, the global interconnection resources may not include single lines 385. Single lines 385 permit flexibility in interconnecting signals and LABs 200, but for many of the logic designs programmed into PLDs, a LAB 200 must be connected to at least one other LAB 200. In view of this, the circuitry and other overhead required to implement single lines may be excessive, leading to greater power consumption and larger integrated circuit die sizes than necessary. Further, certain interconnection resources such as switch boxes 310 (used to programmable couple multiple single lines 385) may become the limiting factor in the size of the design that may be implemented in the PLD. Therefore, an effective, efficient PLD architecture may include double lines 360 and 370, but not single lines 385.

Still further embodiments of the present invention may include triple lines, quadruple lines, quintuple lines, sextuple lines, and other similar interconnection resources. Furthermore, in other embodiments of the present invention, there may be special, direct and indirect, connections between LABs 200 that do not pass through the global interconnection resources.

Switch boxes 310 are used to connect global interconnection resources with other global interconnection resources. A more detailed diagram of a switch box 310 is shown in Fig. 14A. A plurality of GH 210 and GV 220 conductors input into and output from switch box 310. GH 210 and GV 220 conductors represent global interconnection resources. For example, GH 210 and GV 220 may correspond to horizontal and vertical double lines 360 and 370 of Fig. 13.

Switch box 310 is shown as a square arrangement in Fig. 14A, but could be a nonsquare arrangement in other embodiments. Generally, switch box 310 allows a GH 210 or GV 220 conductors to programmably connect to three similar conductors in other directions--in the same direction or to switch directions ("turn a corner") from the horizontal direction to the vertical direction, or vice versa.

Switch boxes 310 are partially populated structures providing selected intersections 400 where GH 210 and GV 220 conductors can be programmably coupled with other such conductors. GH 210 and GV 200 conductors represent global interconnection resources of PLD 121. In particular, GH 210 and GV 220 conductors may represent global interconnection resources including long lines, double lines, single lines, and combinations of these. For example, in the embodiment of Fig. 13, GH 210 and GV 220 conductors represent horizontal double lines 360 and vertical double lines 370, respectively.

Programmable connections are made at selected intersections 400 using multiplexers (not shown). More specifically, GH 210 and GV 220 conductors are programmably connectable at selected intersections 400 of GH 210 and GV 220 conductors. Generally, selected intersections 400 are arbitrarily chosen. In Fig. 14A, selected intersections 400 are arranged diagonally, from an upper left-hand corner of switch box 310 to a lower right-hand corner of switch box 310. However, in other embodiments, selected intersections 400 may be arranged in switch box 310 in another arbitrary arrangement.

In a further embodiment of the present invention, switch boxes 310 contain multiplexers for programmably coupling long lines 370 and 380. Signals routed using long lines 370 and 380 will extend the length and width of PLD 121, as described earlier. Moreover, long lines 370 and 380 may be heavily loaded from driving many LABs 200. Consequently, in one embodiment, the programmable multiplexers of switch boxes 310 for programmably connecting long lines 370 and 380 are buffered to allow signals along these lines to drive the entire length and width of the array. In a still further

embodiment of the present invention, GH 210 and GV 220 conductors may represent a combination of horizontal and vertical long lines 370 and 380, and horizontal and vertical double lines 360 and 370, Moreover, long lines 370 and 380 may be buffered while double lines 360 and 370 are not.

Further details of a selected intersection 400 are shown in Fig. 14B. GH 210A, GH 210B, GV 220A, and GV 220B are global conductors which "intersect" at a various selected intersections 400 in Fig. 14A. As discussed earlier, GH 210A, GH 210B, GV 220A, and GV 220B conductors may represent global interconnection resources including long lines, double lines, single lines, and combinations of these. For example, in the embodiment of Fig. 13, GH 210A, GH 210B, GV 220A, and GV 220B conductors represent horizontal double lines 360 and vertical double lines 370, respectively.

GH 210A, GH 210B, GV 220A, and GV 220B conductors are programmably connectable to another. More specifically, GH 210A is programmably connectable to GV 220A (as indicated by dashed connection 405), GH 210B (as indicated by dashed connection 410), and GV 220B (as indicated by dashed connection 415). GV 220A is programmably connectable to GV 220B (as indicated by dashed connection 420) and GH 210B (as indicated by dashed connection 425). GH 210B is programmably connectable to GV 220B (as indicated by dashed connection 430). These programmable connections are bidirectional and signals can travel in both direction. For example, a signal can be passed along programmable connection 405 from GH 210A to GV 220A and from GV 220A to GH 210A.

Returning to Fig. 13, as discussed earlier, partially populated multiplexer regions 320 are for programmably coupling horizontal and vertical double lines 360 and 370 to horizontal and vertical long lines 340 and 350. In partially populated multiplexer regions 320, long lines 340 and 350 intersect double lines 360 and 370. At these intersections, long lines 340 and 350 may be programmably coupled to double lines 360 and 370. More specifically, regions 320 are implemented using partially populated multiplexer structures. In partially populated multiplexer

structures, long lines 340 and 350 may be programmably coupled to double lines 360 and 370 at selected intersections. In further embodiments of the present invention, regions 320 may be fully populated multiplexer structures, where long lines 340 and 350 may be programmably coupled to double lines 360 and 370 at all intersections. In still further embodiments of the present invention, regions 320 may be half-populated multiplexer structures, where long lines 340 and 350 may be programmably coupled to double lines 360 and 370 at half of the intersections.

Half-populated multiplexer regions 330 are for programmably coupling LAB input-output lines 380 from LABs 200 to horizontal and vertical double lines 360 and 370. In half-populated multiplexer regions 330, LAB input-output lines 380 intersect double lines 360 and 370. At these intersections, LAB input-output lines 380 may be programmably coupled to double lines 360 and 370. More specifically, regions 320 are implemented using half-populated multiplexer structures. In partially populated multiplexer structures, LAB input-output lines 380 may be programmably coupled to double lines 360 and 370 at a selected half of the intersections. In further embodiments of the present invention, regions 330 may be fully populated multiplexer structures, where LAB input-output lines 380 may be programmably coupled to double lines 360 and 370 at all intersections. In a still further embodiment of the present invention, regions 330 may be partially populated multiplexer structures, where LAB input-output regions 380 may be programmably coupled to double lines 360 and 370 at a selected portion of the intersections.

Fig. 15 shows a simplified block diagram of LAB 200 of Figs. 2 and 3. LAB 200 is comprised of a varying number of logic elements (LEs) 500, sometimes referred to as "logic cells," and a local (or internal) interconnect structure 510. LAB 200 has eight LEs 500, but LAB 200 may have any number of LEs, more or less than eight. In a further embodiment of the present invention, LAB 200 has two "banks" of eight LEs for a total of sixteen LEs, where each bank has separate inputs, outputs, control signals, and carry chains.

A more detailed description of LE 500 of the present invention is given below in connection with Fig. 16. A general overview is presented here, sufficient to provide a basic understanding of LAB 200. LE 500 is the smallest logical building block of a PLD. Signals external to the LAB, such as from GHs 210 and GVs 220, are programmably connected to LE 500 through local interconnect structure 510, although LE 500 may be implemented in many architectures other than those shown in Figs. 2 and 3. In one embodiment, LE 500 of the present invention incorporates a function generator that is configurable to provide a logical function of a number of variables, such a four-variable boolean operation. As well as combinatorial functions, LE 500 also provides support for sequential and registered functions using, for example, D flip-flops.

LE 500 provides combinatorial and registered outputs that are connectable to the GHs 210 and GVs 220, outside LAB 200. Furthermore, the outputs from LE 500 may be internally fed back into local interconnect structure 510; through local interconnect structure 510, an output from one LE 500 may be programmably connected to the inputs of other LEs 500, without using the global interconnect structure's GHs 210 and GVs 220. Local interconnect structure 510 allows short-distance interconnection of LEs, without utilizing the limited global resources, GHs 210 and GVs 220. Through local interconnect structure 510 and local feedback, LEs 500 are programmably connectable to form larger, more complex logical functions than can be realized using a single LE 500. Furthermore, because of its reduced size and shorter length, local interconnect structure 510 has reduced parasitics compared to the global interconnection structure. Consequently, local interconnect structure 510 generally allows signals to propagate faster than through the global interconnect structure.

Fig. 16 shows a block diagram of a specific embodiment of LAB 200 of the present invention. LAB 200 of Fig. 16 is configurable to implement logic functions. LAB 200 of Fig. 16 does not have a well-defined LE structure like

LAB 200 of Fig. 15. In contrast, LAB 200 of Fig. 16 is a coarser-grained entity without clearly defined repeated structures such as LEs (which is referred to as a fine-grained entity). Here, LAB 200 has eight "primary" programmable function generators. These primary programmable function generators include "primary" four-input look-up tables (LUTs) 601, 605, 610, 615, 620, 625, 630, and 635 in a first level.

LUTs are programmable elements configurable to provide a logical function. In particular, a four-input LUT is configurable to produce the sixteen possible logical outputs for any boolean operation of the four variables. Instead of a look-up table, LUTs may be designed using other programmable systems for performing and/or functionality such as logic gates, flip-flops, multiplexers, and programmable AND-OR arrays.

In a preferred embodiment, LUTs are implemented using a random access memory (RAM). More specifically, LUTs are implemented using a 16-bit RAM, in one specific embodiment, each bit storing an output state corresponding to one of, *e.g.*, sixteen possible input combinations. In further embodiments of the present invention, LUTs may be implemented using other types of memories besides a RAM, such as a first-in, first-out (FIFO) memory or content-addressable memory (CAM), or a combination of these.

A RAM may be constructed using many different fabrication technologies including fuse, antifuse, ferromagnetic core, erasable programmable read-only memory (EPROM), and electrically erasable programmable read-only memory (EEPROM) technology. A RAM may also be constructed from dynamic random access memory (DRAM) or static random access memory (SRAM) technology. In a preferred embodiment of the present invention, the LUTs of Fig. 16 use SRAM memory.

LUTs 601, 605, 610, 615, 620, 625, 630, and 635 have four inputs, which are for the four variables used to select a particular output for that LUT. LUT 601 has four inputs 638; LUT 605 has four inputs 640; LUT 610 has four inputs 642; LUT 615 has four inputs 644; LUT 620 has four inputs 646; LUT 625 has four inputs 648; LUT 630 has four inputs 650; and

LUT 635 has four inputs 652. These inputs form part of local interconnect structure 510 (described above) and also a portion of LAB input-output lines 380 of Fig. 13. Signals from within and external to LAB 200 may be connected to these inputs. For example, signals from double lines 360 and 370 may be programmably connected to these inputs of LAB 200.

In addition to the primary LUT inputs, the inputs to LAB 200 in local interconnect structure 510 include eight dedicated inputs 654, 656, 658, 660, 662, 664, 666, and 668. A primary four-input LUT is associated with a particular dedicated input. More specifically, dedicated inputs 654, 656, 658, 660, 662, 664, 666, and 668 are associated with primary LABs 601, 605, 610, 615, 620, 625, 630, and 635, respectively. Dedicated inputs 654, 656, 658, 660, 662, 664, 666, and 668 have multiple uses which are described further below.

These dedicated inputs and the inputs to the LUTs of LAB 200 may be programmably coupled to a signal provided on local interconnect structure 510. In one embodiment, local interconnect structure 510 is a half-populated multiplexer structure. In a half-populated multiplexer structure, only half of the provided signals may be coupled to a particular LUT input. In other embodiments, local interconnect structure 510 may be a fully populated or partially populated multiplexer structure. In a fully populated multiplexer structure, every signal may be coupled to every LUT input. In a partially populated multiplexer structure, only a selected portion of the signals may be coupled to a particular LUT input.

Conceptually, LAB 200 of Fig. 16 may be divided into two groupings of LUTs, both groupings having substantially similar configurations and connections between elements. In particular, LUTs 601, 605, 610, and 615 form a first LUT grouping; LUTs 620, 625, 630, and 635 form a second LUT grouping. This description will only discuss the connections for LUTs 601, 605, 610, and 615 in detail, since LUTs 620, 625, 630, and 635 are similarly connected.

In addition to primary LUTs 601, 605, 610, 615, 620, 625, 630, and 635, there are six secondary function generators. These secondary function generators include LUTs, 670, 672, 674, 676, 678, and 680 in a second level. Generally, the outputs signals from the primary LUTs are fed, directly and indirectly, into the inputs of the secondary LUTs so that larger, more complex logical function can be created from the combination of primary and secondary LUTs. Analogous to the two groupings of the primary LUTs, there are two groupings of secondary LUTs. A first grouping of secondary LUTs is associated with the first grouping of primary LUTs. Similarly, a second grouping of secondary LUTs is associated with the second grouping of primary LUTs. The first grouping contains secondary LUTs 670, 672, and 674. The second grouping contains LUTs 676, 678, and 680. This description will only discuss the connections for LUTs 670, 672, and 674 in detail, since LUTs 676, 678, and 680 are similarly connected.

More specifically, for the first grouping of LUTs, regarding secondary two-input LUT 670, an output from primary LUT 601 is directly coupled to one of two inputs to secondary two-input LUT 670. An output from primary LUT 605 is directly coupled to another input of secondary two-input LUT 670. Regarding secondary two-input LUT 674, an output from primary LUT 610 is directly coupled to one of two inputs to secondary two-input LUT 674. An output from primary LUT 615 is directly coupled to another input of secondary two-input LUT 674.

The second grouping of LUTs are similarly connected. Regarding secondary two-input LUT 676, an output from primary LUT 620 is directly coupled to one of two inputs to secondary two-input LUT 676. An output from primary LUT 625 is directly coupled to another input of secondary two-input LUT 676. Regarding secondary two-input LUT 680, an output from primary LUT 630 is directly coupled to one of two inputs to secondary two-input LUT 680. An output from primary LUT 635 is directly coupled to another input of secondary two-input LUT 680.

Secondary two-input LUTs 670, 672, 678, and 680 are used to generate logic functions based on outputs from the

specified primary LUT. These secondary LUTs are used to create larger, more complex logical functions than available with a single primary LUT. In particular, the secondary LUTs facilitate the combination of multiple primary LUTs. For example, secondary two-input LUT 670 can be used to combine primary LUTs 601 and 605 to create a larger five-input LUT for handling functions of up to five variables. Since there are four secondary two-input LUTs, 670, 672, 678, and 680 in the embodiment shown in Fig. 16, four five-input logic functions can be implemented.

LAB 200 of Fig. 16 also includes a plurality of programmable multiplexers 684. Multiplexers 684 are programmably configured to couple a multiplexer input to a multiplexer output. Programmable multiplexers 684 may have an arbitrary number of inputs. In Fig. 16, multiplexers 684 are two-input multiplexers. Multiplexers 684 are controlled, or configured, by way of user-programmable memory cells (not shown), such as SRAM bits. Depending upon the state of such user-programmed bits, an appropriate input of multiplexer 684 is programmably coupled to the output of multiplexer 684.

For the first grouping of LUTs, a multiplexer 684 programmably couples dedicated input 654 and the output of primary LUT 601 to a first input of secondary four-input LUT 672. A multiplexer 684 programmably couples an output of primary LUT 605 and dedicated input 656 to a second input of secondary four-input LUT 672. A multiplexer 684 programmably couples an output of primary LUT 610 and dedicated input 658 to a third input of secondary four-input LUT 672. A multiplexer 684 programmably couples an output of primary LUT 615 and dedicated input 660 to a fourth input of secondary four-input LUT 672.

The second grouping of LUTs are similarly connected to secondary four-input LUT 678. Specifically, a multiplexer 684 programmably couples dedicated input 662 and the output of primary LUT 620 to a first input of secondary four-input LUT 678. A multiplexer 684 programmably couples an output of primary LUT 625 and dedicated input 664 to a second input of secondary four-input LUT 678. A multiplexer 684 programmably

couples an output of primary LUT 630 and dedicated input 666 to a third input of secondary four-input LUT 678. A multiplexer 684 programmably couples an output of primary LUT 635 and dedicated input 668 to a fourth input of secondary four-input LUT 678.

Secondary four-input LUTs 672 and 678 are used tc generate logic functions based on outputs from a combination of primary LUTs and dedicated inputs. These secondary LUTs 672 and 678 are used to create larger, more complex logical functions than available with a single primary LUT. Secondary LUTs 672 and 678 facilitate the combination of multiple primary LUTs. For example, secondary four-input LUTs 672 may be used to combine primary LUTs 601, 605, 610, and 615 to create a larger six-input LUT for handling functions of up to six variables. Since there are two secondary four-input LUTs, 672 and 678, in the embodiment shown in Fig. 16, two six-input logic functions can be implemented.

Therefore, in LAB 200 of Fig. 16, two six-input logic functions and four five-input logic functions (see above), and combinations of these, can be implemented. For example, LAB 200 of Fig. 16 has eight four-input LUTs, 601, 605, 610, 615, 620, 625, 630, and 635; another two four-input LUTs can be implemented using dedicated inputs 654, 656, 658, 660, 662, 664, 666, and 668, and secondary four-inputs LUTs 672 and 678. In particular, multiplexers 684 are configured to programmably couple dedicated inputs 654, 656, 658, and 660 to secondary four-input LUT 672; and dedicated inputs 662, 664, 666, and 668 are programmably coupled to secondary four-input LUT 678. In this configuration, ten four-input LUTs are available for use.

Primary four-input LUTs 601, 605, 610, 615, 620, 625, 630, and 635 have combinatorial path outputs 687 and a registered path outputs 689. LAB 200 of the present invention has eight combinatorial outputs 687 and eight registered outputs 689. Combinatorial path outputs 687 are used to output results of combinatorial logic functions which depend on the present input states in some predetermined fashion; in

Fig. 16, this is governed by the configuration information within the LUTs. Registered path outputs 689 are connected to storage blocks 691; these outputs 689 are used to output of registered or sequential logic functions which depend on both the input states and the previous history. Registered (or sequential) functions are implemented using some form of memory circuit, including circuits such as registers, flip-flops, and the like.

Combinatorial outputs 687 are programmably selected using programmable multiplexers 684. For the first grouping of LUTs, a multiplexer 684 programmably couples the output of primary LUT 601 and an output of secondary LUT 670 to a combinatorial output 687. A multiplexer 684 programmably couples the output of primary LUT 605 and an output of secondary LUT 672 to a combinatorial output 687. A multiplexer 684 programmably couples the output of primary LUT 610 and the output of secondary LUT 672 to a combinatorial output 687. A multiplexer 684 programmably couples the output of primary LUT 615 and an output of secondary LUT 674 to a combinatorial output 687.

Similarly, for the second grouping of LUTs, a multiplexer 684 programmably couples the output of primary LUT 620 and an output of secondary LUT 676 to a combinatorial output 687. A multiplexer 684 programmably couples the output of primary LUT 625 and an output of secondary LUT 678 to a combinatorial output 687. A multiplexer 684 programmably couples the output of primary LUT 630 and the output of secondary LUT 678 to a combinatorial output 687. A multiplexer 684 programmably couples the output of primary LUT 635 and an output of secondary LUT 680 to a combinatorial output 687.

Combinatorial outputs 687 form a portion of LAB input-output lines 380 of Fig. 13 and are programmably connectable to the global interconnect structure, including long lines and double lines. Furthermore, as discussed earlier, in one embodiment of the present invention, combinatorial outputs 687 are programmably connectable, directly, to horizontal and vertical long lines 360 and 370.

Moreover, combinatorial outputs 687 may be programmably connected through the global interconnect structure to LAB input-output lines 380 inputting into other LABs 200 or the same LAB 200 to form more complex logical functions from a combination of LABs 200.

In the embodiment shown in Fig. 16, combinatorial outputs 687 of the present invention feedback into local interconnect structure 510 (not shown to simplify the drawing). As discussed earlier, local interconnect structure 510 is a fully, partially, or half-populated multiplexer region that allows coupling of these combinatorial outputs 687 to the inputs. Consequently, via local interconnect structure 510, combinatorial outputs 687 may be programmably coupled to inputs and dedicated inputs of the primary LUTs, without using interconnect resources outside the LAB such as global interconnect conductors.

In LAB 200 of Fig. 16, there are eight storage blocks 691. A primary four-input LUT may be programmably coupled to a storage block 691 for providing a registered output 689. In particular, for the first grouping of LUTs, a data input of storage block 691 may be programmably coupled to signals from dedicated input 654, the output of primary LUT 601, and the output of secondary LUT 670. More specifically, a multiplexer 684 programmably couples to this data input of storage block 691 signals from: dedicated input 654 and the output of another multiplexer 684 (discussed earlier as being coupled to combinatorial output 687), which programmably selects between the output of primary LUT 601 and the output of secondary LUT 670. These configuration paths could have been obtained using other circuitry such as a three-input multiplexer. However, two two-input multiplexers 684 were used in the embodiment for Fig. 16 since one multiplexer 684 is used for combinatorial output 687. This is similarly the case for the other storage blocks 691.

Further, a data input of a storage block 691 may be programmably coupled to dedicated input 656, the output of primary LUT 605, and the output of secondary LUT 672. In particular, a multiplexer 684 programmably couples to this

data input of storage block 691 signals from: dedicated input 656 and the output of another multiplexer 684 (discussed earlier as being coupled to combinatorial output 687), which programmably selects between the output of primary LUT 605 and the output of secondary LUT 672. A data input of a storage block 691 may be programmably coupled to dedicated input 658, the output of primary LUT 610, and the output of secondary LUT 672. In particular, a multiplexer 684 programmably couples to this data input of storage block 691 signals from: dedicated input 658 and the output of another multiplexer 684 (discussed earlier as being coupled to combinatorial output 687), which programmably selects between the output of primary LUT 610 and the output of secondary LUT 672. A data input of a storage block 691 may be programmably coupled to dedicated input 660, the output of primary LUT 615, and the output of secondary LUT 674. In particular, a multiplexer 684 programmably couples to this data input of storage block 691 signals from: dedicated input 660 and the output of another multiplexer 684 (discussed earlier as being coupled to combinatorial output 687), which programmably selects between the output of primary LUT 615 and the output of secondary LUT 674.

Similarly, for the second grouping of LUTs, a data input of a storage block 691 may be programmably coupled to dedicated input 662, the output of primary LUT 620, and the output of secondary LUT 676. In particular, a multiplexer 684 programmably couples to this data input of storage block 691 signals from: dedicated input 662 and the output of another multiplexer 684 (discussed earlier as being coupled to combinatorial output 687), which programmably selects between the output of primary LUT 620 and the output of secondary LUT 676. A data input of a storage block 691 may be programmably coupled to dedicated input 664, the output of primary LUT 625, and the output of secondary LUT 678. In particular, a multiplexer 684 programmably couples to this data input of storage block 691 signals from: dedicated input 664 and the output of another multiplexer 684 (discussed earlier as being coupled to combinatorial output 687), which

programmably selects between the output of primary LUT 625 and the output of secondary LUT 678. A data input of a storage block 691 may be programmably coupled to dedicated input 666, the output of primary LUT 630, and the output of secondary LUT 678. In particular, a multiplexer 684 programmably couples to this data input of storage block 691 signals from: dedicated input 666 and the output of another multiplexer 684 (discussed earlier as being coupled to combinatorial output 687), which programmably selects between the output of primary LUT 630 and the output of secondary LUT 678. A data input of a storage block 691 may be programmably coupled to dedicated input 666, the output of primary LUT 635, and the output of secondary LUT 680. In particular, a multiplexer 684 programmably couples to this data input of storage block 691 signals from: dedicated input 668 and the output of another multiplexer 684 (discussed earlier as being coupled to combinatorial output 687), which programmably selects between the output of primary LUT 635 and the output of secondary LUT 680.

Storage blocks 691 are used to store a logic state. Many different logical components elements can be used to form storage blocks 691 including, among others, memory cells, D, T, S-R, J-K, and other types of latches and registers. For example, in the embodiment shown in Fig. 16, storage blocks 691 are D-type registers. In other embodiments of the present invention, LAB 200 may contain T, S-R, J-K, and other types of latches and registers, and combinations of these. Furthermore, in another embodiment, storage block 691 is programmably configurable to operate also as a transparent latch.

LAB 200 has CLK0 693, CLK1 694, CE 696, *S* 697, *R* 698, and DIN 699 input lines. These lines govern the functionality, which are sometimes referred to as the "secondary functions," of storage block 691. These lines form a portion of LAB input-output lines 380 (described above), which may be programmably connected to via the global interconnection resources, which include switch boxes 310, double lines 360 and 370, and long lines 340 and 350.

In typical operation, storage block 691 latches in data from its data input and outputs data at its output 689 in response to a clock signal input. A multiplexer 684 programmably couples a CLK0 693 signal and a CLK1 694 signal to the clock signal input of storage block 691. The embodiment shown in Fig. 16 has eight of these multiplexers 684, which are coupled to the clock signal inputs of storage blocks 691, one multiplexer 684 for a storage block 691. Depending on how multiplexer 684 is configured, the clock signal input of storage block 691 can be controlled by either CLK0 693 or CLK1 694 signals. Furthermore, since multiplexers 684 can be programmably configured independently for the eight storage blocks 691, a portion of the registers in LAB 200 may be controlled by CLK0 693 while the other portion is controlled by CLK1 694. Eight storage blocks 691 may be also controlled by the same CLK0 693 signal or CLK1 694 signal.

Storage block 691 may operate in various modes including, among others, leading-edge-triggered register, trailing-edge-triggered register, active-high latch, active-low latch, or as a direct combinatorial bypass. More specifically, in leading-edge-triggered register mode, storage block 691 will function as a register, latching in new data and outputting stored data in response to a leading edge of a clock input. In trailing-edge-triggered register mode, storage block 691 will function as a register that will is responsive to a falling edge of a clock input. In active-high latch mode, storage block 691 will function as a latch, latching data which its clock input is a high. In active-low latch mode, storage block 691 will function as a latch, latching data when its clock input is a low. In direct combinatorial bypass mode, storage block 691 will pass data through without any clocking; in this mode, storage block 691 becomes transparent. Storage blocks 691, or portion thereof, may operate in these modes or a combination of these modes. For example, in the embodiment shown in Fig. 16, storage blocks 691 operate in substantially the same mode, as leading-edge-triggered, D-type registers. In another example, one portion of the storage blocks 691 may operate in leading-edge-

triggered register mode and another portion operates in active-high latch mode.

Furthermore, storage blocks 691 have clock enable (CE), set (*S*), and reset (*R*) signal inputs. In the embodiment shown in Fig. 16, a CE 696 signal directly couples to the CE signal input of storage blocks 691. CE 696 enables clocking of storage block 691, in response to a signal at the clock signal input. More specifically, in operation, when CE 696 is asserted, the clock signal input to storage block 691 may be used to clock storage block 691; when CE 696 is not asserted, storage block 691 may not be clocked. As discussed earlier, CE 696 may be programmably coupled to the global interconnection resources, outside LAB 200. Furthermore, in other embodiments of the present invention, CE 696 may, for example, come from multiple sources and is programmably coupled through a multiplexer 684.

An *S* 697 signal directly couples to the *S* signal input of storage blocks 691. *S* 697 "sets" (or presets) the storage block 691, loading storage block 691 with a "1" or logic high. More specifically, in operation, when *S* 697 is asserted, storage block 691 is asynchronously set (*i.e.*, sets storage block 691 regardless of the current state of storage block 691); when *S* 697 is not asserted, normal operation of storage block 691 is enabled. In other embodiments of the present invention, *S* 697 may synchronously set storage block 691, which means storage block 691 is set at the next clock pulse. As discussed earlier, *S* 697 may be programmably coupled to the global interconnection resources, outside LAB 200. Furthermore, in other embodiments of the present invention, *S* 697 may, for example, come from multiple sources and is programmably coupled through a multiplexer 684.

An *R* 698 signal directly couples to the *R* signal input of storage blocks 691. *R* "resets" (or clears) storage block 691, loading the storage block 691 with a "0" or logic low. More specifically, in operation, when *R* 698 is asserted, storage block 691 is asynchronously reset (*i.e.*, resets storage block 691 regardless of the current state of storage block 691); when *R* 698 is not asserted, normal operation of

storage block 691 is enabled. In other embodiments of the present invention, *R* 698 may synchronously set storage block 691, which means storage block 691 is reset at the next clock pulse. As discussed earlier, *R* 698 may be programmably coupled to the global interconnection resources, outside LAB 200. Furthermore, in other embodiments of the present invention, *R* 698 may, for example, come from multiple sources and is programmably coupled through a multiplexer 684. In Fig. 16, the *S* 697 and *R* 698 signals are separate signals. In other embodiments, however, the *S* 697 and *R* 698 signals and corresponding functions may be implemented using a single, combined control line.

A DIN 699 signal is also shown in Fig. 16. However, DIN 699 is not used in the embodiment in Fig. 16. Hence, DIN 699 is not shown coupled to storage block 691. DIN 699 is used in a further embodiment or the present invention described below.

Registered outputs 689 form a portion of LAB input-output lines 380 of Fig. 13 and are programmably connectable to the global interconnect structure, including long lines and double lines. Moreover, as discussed earlier, in one embodiment of the present invention, registered outputs 689 are programmably connectable, directly, to horizontal and vertical long lines 360 and 370. Furthermore, registered outputs 689 may be programmably connected through the global interconnect structure to LAB input-output lines 380 inputting into other LABs 200 or the same LAB 200 to form more complex logical functions from a combination of LABs 200.

In one embodiment of the present invention, registered outputs 689 do not feedback into local interconnect structure 687 as do combinatorial outputs 687. However, registered outputs 689 may be fed back and programmably coupled to inputs of the same LAB 200 via the global interconnect structure (described above). The lack of an "internal" this feedback path saves valuable integrated circuit area for other features. However, in other embodiments of the present invention, registered outputs 689 may internally feedback into local interconnect structure 687,

and may be programmably coupled to the primary LUTs as described above.

Fig. 17 is a block diagram showing a carry chain scheme for a LAB 200 of Fig. 16. Circuit features shown Fig. 16 are in the embodiment shown in Fig. 17. However, in order to simplify Fig. 17, some of the LAB circuitry has been omitted from the drawing, specifically, the secondary LUTs, some multiplexers 684, and some connections.

When implementing a counter, adder, or other similar function, primary LUTs of LAB 200 are configured to implement a single bit of the logic function. Also, to implement these types of logic function, a carry chain is typically necessary. A carry chain may be implemented when a carry mode is enabled in LAB 200. Among other uses, carry chains may be used to implement arithmetic logic functions such as adders, accumulators, and counters.

A carry-in (CIN) signal is input into LAB 200 via DIN 699. CIN is coupled to the DIN 699 input described earlier; this is a use for the DIN 699 input. Also, as discussed earlier, DIN 699 is programmably connectable to the global interconnection resources; therefore, a CIN signal may be programmably coupled to DIN 699 via long lines, double lines, and other similar conductors and resources. For example, CIN may be output from one LAB 200 and programmably connected through double lines 360 and 370 to another LAB 200.

In a carry chain, primary LUT 635, which is coupled to DIN 699, will represent a first bit for a counter, adder, or other similar logic function implemented. The counter or similar function may have an arbitrary size. Within a single LAB 200, a counter or other similar logical function can have a word size from one bit up to eight bits. The last bit in the carry chain outputs a carry-out (COUT) 705 signal. COUT 705 is programmably connectable to the global interconnection resources. A second bit in the carry chain would be represented by primary LUT 630. A third bit in the carry chain would be represented by primary LUT 625. Similarly, the fourth, fifth, sixth, seventh, and eighth bits

for the logic function would be represented by LUTs 620, 615, 610, 605, and 601, respectively.

If more than eight bits are needed, a wider logic function and carry chain may be implemented by programmably coupling multiple LABs 200 through the global interconnect. For example, CIN would input through DIN 699 of one LAB 200 and the carry chain would propagate through the eight LUTs to COUT. This COUT will be programmably coupled through the global interconnect to a DIN 699 of another LAB 200. Multiple LABs 200 are programmably combined in this fashion until the desired-size logic function is obtained.

Furthermore, to implement a counter or similar logic function, an *A* 710 input and a *B* 720 input are data inputs to the primary LUTs of a LAB 200. *A* 710 and *B* 720 are two operands which will be operated on by LAB 200. For example, in an adder, *A* 710 will be one addend and *B* 720 is the other addend. An *A* 710 input for a particular primary LUT is a particular bit of that operand. A *B* 720 input for a particular primary LUT is a particular bit of that operand. Another input, U/D 725, to LUT 405 is used for an up-down count control when implementing an up-down counter. For example, for an up-down counter, when U/D 725 is asserted, the counter will increment when clocked; when U/D 725 is not asserted, the counter will decremented when clocked.

An output of the primary LUTs (configured to implement a counter-type logic function in Fig. 17) is a SUM signal. SUM is the result of the logical operation on the *A* 409 and *B* 413 operands. Carry logic blocks 740 contain the logic for implementing the carry scheme of the present invention. A carry logic block 740 is coupled to a primary LUT in LAB 200. LAB 200 has eight carry logic blocks 740.

More specifically, carry logic blocks 740 are coupled to the *A* 710, *B* 720, and U/D 725 inputs of a primary LUT. A first carry logic block 425, coupled to the inputs of LUT 635, takes a carry input from DIN 699. Other carry logic blocks 740 take a carry input from a previous carry logic block 425 (as indicated by connection 750). A last carry logic block in the chain (shown coupled to primary LUT 601 in

Fig. 17) generates the COUT 705 signal, which may be coupled to other LABs 200 through the global interconnection resources. Based on its *A* 710, *B* 720, and U/D 725 inputs, carry logic block 425 determines whether to generate or propagate a carry signal to the next carry logic block (via connection 750) in the chain.

In one embodiment, the carry chain of the present invention employs a carry propagate-generate technique. More specifically, a carry bit is either propagated or generated by a primary LUT and passed onto to a succeeding primary LUT. Using this technique, an entire carry chain implemented can be within a LAB 200. The carry chain implementation of the present invention provides easier and more efficient implementation of some logic functions such as an up-down counter.

When the carry chain feature is used, secondary LUTs 672 and 678 (shown in Fig. 16) may still be accessed by the dedicated inputs (*i.e.*, 654, 656, 658, 660, 662, 664, 666, and 668). Secondary LUTs 672 and 678 may drive registers not used by other LUTs. Furthermore, if some primary LUTs are not used in the carry mode, these primary LUTs can be used for implementing logic functions, without impacting the carry chain. For example, three primary LUTs (*e.g.*, 635, 630, and 625) may be used in carry mode; the remaining five primary LUTs (*i.e.*, 620, 615, 610, 605, and 601) may be used to implement other logical function, without affecting the functionality of the carry chain.

Fig. 18 is a block diagram of LAB 200 of Fig. 16 configured to implement a random access memory (RAM). Circuit features shown Fig. 16 are in the embodiment shown in Fig. 18. However, in order to simplify Fig. 18, some of the LAB circuitry has been omitted from the drawing.

As shown in Fig. 18, LAB 200 implements a RAM memory in a RAM mode. In the RAM mode, the primary LUTs serve as individual RAM blocks for storing data. A primary four-input LUT can store up to 16 bits of data. Since there are eight primary LUTs per LAB, LAB 200 can store 128 bits, or 16 bytes of data. The RAM may be organized in a format having any

width and depth size. For example, the RAM may have a 4-bit word size and thirty-two words. Furthermore, multiple LABs 200 may be programmably combined to implement larger sized RAMs.

In the RAM mode, the inputs to the primary LUTs are the address pins for the RAM. The CE 696 input (described above) serves as input for a write enable (WE) control signal. WE enables writing or reading of the RAM cells. When WE is high, the RAM may be written; when WE is low, the RAM may be read, but not written. When implementing larger RAMs using multiple LABs, in order to reduce skew between LABs, a dedicated (global) clock network or long lines 340 and 350 may be used to conduct the WE signal within PLD 121. Furthermore, LAB 200 has a program register 810 for allowing the synchronization of the WE signal with either dedicated clock signal CLK0 or CLK1; the resulting synchronized signal, WES 820, is passed to a WE decode logic block 830.

A primary LUT has a WE decode logic block 830. LAB 200 has eight WE decode logic blocks 830, one coupled to one primary LUT. WE decode logic block 830 enables writing or reading of the RAM based on the state of WES 820. Furthermore, a RAM bit 835 is used by WE decode logic block 830 to control the organization of the RAM. When RAM bit 835 is programmed, the RAM of LAB 200 has a particular the word width and depth size such as 128 bits by 1. When RAM bit 835 is not programmed, the RAM of LAB 200 may be organized as 16 bits by 8 bits.

To store or write data into a primary LUT, a data-in multiplexer 684 programmably selects a source of data from either a dedicated input pin or the DIN 699 input. For example, data from dedicated input pin 654 may be written into primary LUT 601; data from dedicated input pin 660 may be written into primary LUT 615. In the alternative, data from DIN 699 may be written into primary LUTs 601 and 615.

To retrieve or read data from a primary LUT, data from a primary LUT is output through the respective LUT output. A multiplexer 684 programmably couples a dedicated input and the output of a primary LUT to an input of a

secondary four-input LUT. More specifically, in the first grouping of LUTs, a multiplexer 684 programmably couples dedicated input 654 and the output of a primary LUT 601 to a first input of secondary four-input LUT 672. A multiplexer 684 programmably couples dedicated input 656 and the output of a primary LUT 605 to a second input of secondary four-input LUT 672. A multiplexer 684 programmably couples dedicated input 658 and the output of a primary LUT 610 to a third input of secondary four-input LUT 672. A multiplexer 684 programmably couples dedicated input 660 and the output of a primary LUT 615 to a fourth input of secondary four-input LUT 672.

Similarly for the second grouping of LUTs, a multiplexer 684 programmably couples dedicated input 662 and the output of a primary LUT 620 to a first input of secondary four-input LUT 678. A multiplexer 684 programmably couples dedicated input 664 and the output of a primary LUT 625 to a second input of secondary four-input LUT 678. A multiplexer 684 programmably couples dedicated input 666 and the output of a primary LUT 630 to a third input of secondary four-input LUT 678. A multiplexer 684 programmably couples dedicated input 668 and the output of a primary LUT 635 to a fourth input of secondary four-input LUT 672.

This outputs of secondary LUTs 672 and 678 may be output from LAB 200 using combinatorial output 687 or registered output 689 (connections not shown) as discussed earlier. Secondary LUTs 672 and 678 may perform logical operations on the output signals of the primary LUTs before being output from LAB 200. Furthermore, secondary LUTs 672 and 678 may also be used for decoding addresses for the RAM blocks.

Fig. 19 shows a block diagram of a digital system within which the present invention may be embodied. In the particular embodiment of Fig. 19, a processing unit 101 is coupled to a memory 105 and an I/O 111 and incorporates a programmable logic device (PLD) 121. PLD 121 may be specially coupled to memory 105 through connection 131 and to I/O 111 through connection 135. The system may be a programmed

digital computer system, digital signal processing system, specialized digital switching network, or other processing system. Moreover, the system may be a general purpose computer, a special purpose computer (such as telecommunications equipment) optimized for an application-specific task such as programming PLD 121, or a combination of a general purpose computer and auxiliary special purpose hardware.

Processing unit 101 may direct data to an appropriate system component for processing or storage, execute a program stored in memory 105 or input using I/O 111, or other similar function. Processing unit 101 may be a central processing unit (CPU), microprocessor, floating point coprocessor, graphics coprocessor, hardware controller, microcontroller, programmable logic device programmed for use as a controller, or other processing unit. Furthermore, in many embodiments, there is often no need for a CPU. For example, instead of a CPU, one or more PLDs 121 may control the logical operations of the system.

In some embodiments, processing unit 101 may even be a computer system. In one embodiment, source code may be stored in memory 105, compiled into machine language, and executed by processing unit 101. Processing unit 101 need not contain a CPU and in one embodiment, instructions may be executed by one or more PLDs 121. Instead of storing source code in memory 105, only the machine language representation of the source code, without the source code, may be stored in memory 105 for execution by processing unit 101. Memory 105 may be a random access memory (RAM), read only memory (ROM), fixed or flexible disk media, PC Card flash disk memory, tape, or any other storage retrieval means, or any combination of these storage retrieval means.

Processing unit 101 uses I/O 111 to provide an input and output path for user interaction. For example, a user may input logical functions to be programmed into programmable logic device 121. I/O 111 may be a keyboard, mouse, track ball, digitizing tablet, text or graphical display, touch screen, pen tablet, printer, or other input or output means,

or any combination of these means. I/O 111 may also be a connection or bus to another processing system that provides input or receives output from 101. In one embodiment, I/O 111 includes a printer used for printing a hard copy of any processing unit 101 output. In particular, using I/O 111, a user may print a copy of a document prepared using a word processing program executed using processing unit 101. In other cases, a user may print out a copy of the source code or a listing of the logical functions contained within PLD 121.

PLD may serve many different purposes within the system in Fig. 19. PLD 121 may be a logical building block of processing unit 101, supporting its internal and external operations. PLD 121 is programmed to implement the logical functions necessary to carry on its particular role in system operation.

As some examples of the multitude of uses for PLD 121, processing unit 101 may use PLD 121, through connection 131, to decode memory or port addresses for accessing memory 105 or I/O 111. PLD 121 may be programmed to store data like a memory or specialized memory, where this comes from processing unit 101 or memory 105 (via connection 131). PLD 121 may be used as a microcontroller for a memory 105 device such as a fixed or flexible disk drive. PLD 121 may also be configured to be a microcontroller for an I/O 111 device such as a keyboard or scanner, passing data through connection 135.

In other embodiments, PLD 121 may be used as a controller or specialized processing unit such as a coprocessor for performing mathematical or graphical calculations. In still other embodiments, PLD 121 may be used for telecommunications applications. For example, processing unit 101 would direct data to PLD 121; PLD 121 processes this data; then PLD 121 returns the results to processing unit 101. Furthermore, processing unit 101 may pass or direct a program stored in memory 105 or input using I/O 111 to PLD 121 for execution. These are some of multitude of uses of PLD 121 within a digital system. Also, a system such as the one shown

in Fig. 19 may embody a plurality of PLDs 121, each performing different system functions.

The system shown in Fig. 19 may also be used for programming PLD 121 with a particular logic pattern. A computer program for designing functions into a PLD may be stored in memory 105 and executed using processing unit 101. Then, a design characteristic which is to be programmed into PLD 121 is input via I/O 111 and processed by processing unit 101. In the end, processing unit 101 transfers and programs the design characteristic into PLD 121.

In Fig. 19, processing unit 101 is shown incorporating PLD 121. However, in other embodiments, PLD 121 may be external to processing unit 101, and a PLD interface may be coupled between processing unit 101 and PLD 121. The PLD interface would provide the proper adapters or sockets for interfacing PLD 121 to processing unit 101. Moreover, the PLD interface would provide the proper voltages and electrical characteristics for coupling PLD 121 to processing unit 101.

The invention has now been explained with reference to specific embodiments. However, a number of variations to the invention will be obvious to anyone with skill in the art. For example, the invention need not be limited to circuits that are commonly thought of as PLDs; other types of configurable counter circuits or adder circuits may employ the invention. Also, the particular design of the LABs is not determinative. The invention may be used with SOP-based LABs as well as LUT-based LABs. The invention may also be employed in PLDs having a wide variety of LAB and LE architectures and a number of different interconnect strategies. Additionally, the particular configurations of the global and local interconnect are not determinative. Circuits in which the invention may be employed may employ either a more limited or more extensive global interconnect than those depicted in the specific embodiments. It should also be noted that the device output pins may be configured on different devices as bidirectional input/output pins or as unidirectional output pins. Where the term output pin is used in this application, it is to be understood that an

input/output is included as well. Finally, systems incorporating the invention may be any type of information processing system or subsystem. Finally, the particular configuration of the direct carry and the global interconnect are not determinative. Circuits in which the invention may be employed may have a limited or modified global interconnect and may employ either a more limited or more extensive direct carry chain that those depicted in the specific embodiments. In addition, it will be clear that a PLD may include all of the enhanced elements herein described or may include combinations of the enhancements. It is therefore intended that the invention not be limited except as specified in the attached claims.

WHAT IS CLAIMED IS:

1. A configurable electronic device comprising:

a plurality of logic blocks, at least one logic block comprising outputs and an array of local interconnection lines within said block, said at least one logic block performing selected logical operations said at least one logic block further comprising:

a carry input line; and

a selector with its output coupled to the carry input line, one of its selectable inputs coupled to a direct carry output of an adjacent logic block and another of its selectable inputs coupled to said array of local interconnection lines; ;

a plurality of logic elements, at least one logic element comprising a plurality of outputs and a plurality of inputs wherein said outputs are a logical function of said inputs and wherein at least one logic element has associated with it:

a cascade logic gate for combining an input directed to said logic element with an output of another logic element; and

a cascade connector for conveying a signal from an output of said another logic element to said cascade logic gate;

a general interconnect structure programmably connecting the outputs of one of said logic blocks to said array of local interconnection lines of another of said logic blocks;

an output pin interconnect line, said output pin interconnect line programmably connected to said internal interconnection lines and connected directly to an output pin of said configurable electronic device; and

a wide-input AND gate within at least one of said logic blocks, said wide-input AND gate having inputs connectable to a plurality of outputs of logic elements located within said at least one logic block and having an output that is a logical function of its inputs.

2. The configurable electronic device according to claim 1 wherein at least one logic element further comprises at least two look-up tables said look-up tables having inputs selectively connectable to said inputs of said logic element and having outputs selectively connectable to said output of said logic element.

3. A configurable electronic device comprising:

a plurality of logic elements, at least one logic element comprising a plurality of outputs and a plurality of inputs wherein said outputs are a logical function of said inputs and wherein at least one logic element has associated with it:

a cascade logic gate for combining an input directed to said logic element with an output of another logic element; and

a cascade connector for conveying a signal from an output of said another logic element to said cascade logic gate.

4. The configurable electronic device according to claim 3 wherein a plurality of said logic elements are grouped into a plurality of logic array blocks (LABs), at least one LAB comprising a plurality of inputs and outputs and capable of performing a logic function such that said at least one LAB's outputs represent a logical function of its inputs and further comprising:

a general interconnect structure capable of routing the outputs of one of said LABs to the inputs of another of said LABs; and

at least one LAB-based interconnect, each LAB-based interconnect associated with a particular LAB capable of connecting selected lines of said general interconnect to selected input lines of said LAB.

5. The configurable electronic device as recited in claim 4 further comprising a LAB cascade input line,

selectably connectable to a LAB cascade output of an adjacent LAB or to said global interconnect.

6. A programmable logic device as recited in claim 4 wherein said LAB further comprises a cascade output line and a line for connecting a cascade output to said general interconnect structure.

7. A configurable logic device comprising:

a plurality of logic array blocks (LABs), arranged in a plurality of rows and columns, at least one LAB having a plurality of inputs and outputs and capable of performing a logic function such that said at least one LAB's outputs represent a logical function of its inputs, said logic function being programmable during chip manufacture or use;

a general interconnect structure comprising a plurality of rows of conductors and a plurality of columns of conductors, said interconnect structure capable of routing select signals between said columns of conductors and said rows of conductors;

LAB output lines for selectively connecting said LAB outputs to said general interconnect structure;

at least one LAB-based interconnect associated with a particular LAB capable of connecting selected lines of said general interconnect to selected input lines of said LAB; and

a plurality of logic elements wherein at least one logic element has associated with it:

a cascade logic gate for combining an input directed to said logic element with an output of another logic element; and

a cascade connector for conveying a signal from an output of said another logic element to said cascade logic gate.

8. A configurable electronic device comprising:

a plurality of logic blocks, at least one logic block comprising outputs and an array of local interconnection

lines within said block, said at least one logic block performing selected logical operations;

a general interconnect structure programmably connecting the outputs of one of said logic blocks to said array of local interconnection lines of another of said logic blocks;

wherein said at least one logic block further comprises:

a carry input line; and

a selector with its output coupled to the carry input line, one of its selectable inputs coupled to a direct carry output of an adjacent logic block and another of its selectable inputs coupled to said array of local interconnection lines.

9. The configurable electronic device as recited in claim 8 wherein said at least one logic block further comprises a carry output line connectable to said general interconnect structure.

10. The configurable electronic device as recited in claim 8 wherein said array of local interconnection lines is fully populated to inputs of individual logic elements in said logic block and to said input of said carry-in selector.

11. A programmable logic device comprising:

a plurality of logic array blocks (LABs), at least one LAB comprising a plurality of inputs and outputs and capable of performing a logic function such that said at least one LAB's outputs represent a logical function of its inputs;

a general interconnect structure capable of routing the outputs of one of said LABs to the inputs of another of said LABs;

at least one LAB-based interconnect, each LAB-based interconnect associated with a particular LAB capable of connecting selected lines of said general interconnect to selected input lines of said LAB;

wherein said at least one LAB-based interconnect associated with at least one LAB includes a LAB-based carry input line selectably connectable to the carry-in of said LAB.

12. A programmable logic device as recited in claim 11 wherein said LAB further comprises a direct carry output line and a line for connecting a carry output to said general interconnect structure.

13. A configurable logic device comprising:

a plurality of logic array blocks (LABs), arranged in a plurality of rows and columns, at least one LAB having a plurality of inputs and outputs and capable of performing a logic function such that said at least one LAB's outputs represent a logical function of its inputs, said logic function being programmable during chip manufacture or use;

a general interconnect structure comprising a plurality of rows of conductors and a plurality of columns of conductors, said interconnect structure capable of routing select signals between said columns of conductors and said rows of conductors;

LAB output lines for selectively connecting said LAB outputs to said general interconnect structure;

LAB carry_out lines for selectively connecting a LAB's carry to said general interconnect structure; and

at least one LAB-based interconnect associated with a particular LAB capable of connecting selected lines of said general interconnect to selected input lines of said LAB;

wherein said at least one LAB-based interconnect associated with at least one LAB includes a plurality of LAB input carry_in lines connectable to a LAB's carry input.

14. A logic array block (LAB) for use in a configurable logic device comprising:

a plurality of LAB-input lines for receiving signals;

a plurality of logic elements having an output and having a plurality of login element input lines, said login

element input lines being programmably connectable with said LAB-input lines; and

a LAB input carry line, said LAB input carry line programmably connectable with said LAB-input lines and being selectably connectable to a LAB carry input.

15. The apparatus according to claim 14 wherein said LAB input carry line is programmably connectable to said LAB-input lines through an output line multiplexer, said output line multiplexer controlled by a programmable store.

16. The apparatus according to claim 15 further comprising a carry_out line connectable to said general interconnect structure.

17. A configurable electronic device comprising:

a plurality of logic blocks, at least one logic block comprising outputs to other logic blocks, an array of internal interconnection lines within said block, said at least one logic block performing selected logical operations;

a general interconnect structure programmably connecting the outputs of said logic blocks to said array of internal interconnection lines of another of said logic blocks; and

an output pin interconnect line, said output pin interconnect line programmably connected to said internal interconnection lines and connected directly to an output pin of said configurable logic device.

18. The configurable electronic device as recited in claim 17 further comprising an output driver on said output pin interconnect line.

19. The configurable electronic device as recited in claim 17 wherein said array of internal interconnection lines is fully populated to inputs of individual logic elements in said logic block and to said output pin interconnect line.

20. A programmable logic device comprising:

a plurality of logic array blocks (LABs), at least one LAB comprising a plurality of inputs and outputs and capable of performing a logic function such that said at least one LAB's outputs represent a logical function of its inputs;

a general interconnect structure capable of routing the outputs of one of said LABs to the inputs of another of said LABs;

at least one LAB-based interconnect, each LAB-based interconnect associated with a particular LAB capable of connecting selected lines of said general interconnect to selected input lines of said LAB; and

wherein said at least one LAB-based interconnect associated with at least one LAB includes a plurality of output lines connectable to a device I/O pin.

21. A configurable logic device comprising:

a plurality of logic array blocks (LABs), arranged in a plurality of rows and columns, at least one LAB having a plurality of inputs and outputs and capable of performing a logic function such that said at least one LAB's outputs represent a logical function of its inputs, said logic function being programmable during chip manufacture or use;

a general interconnect structure comprising a plurality of rows of conductors and a plurality of columns of conductors, said interconnect structure capable of routing select signals between said columns of conductors and said rows of conductors;

LAB output lines for selectively connecting said LAB outputs to said rows or said columns of conductors;

at least one LAB-based input interconnect associated with a particular LAB capable of connecting selected lines of said general interconnect to selected input lines of said LAB;

wherein said at least one LAB-based interconnect associated with at least one LAB includes a plurality of output lines connectable to a device I/O pin.

22. A logic array block (LAB) for use in a configurable logic device comprising:

a plurality of LAB-input lines for receiving signals;

a plurality of logic elements having an output and having a plurality of login element input lines, said login element input lines being programmably connectable with said LAB-input lines; and

a plurality of device-output lines, said device-output lines being programmably connectable with said LAB-input lines.

23. A configurable electronic device comprising:

a plurality of logic elements grouped into a plurality of logic blocks;

a plurality of shared input signals connectable to a plurality of logic elements in at least one logic block; and

at least one logic element comprising:

a plurality of outputs and a plurality of inputs wherein said outputs are a logical function of said inputs; and

at least two look-up tables said look-up tables having inputs selectively connectable to said inputs of said logic element and having outputs selectively connectable to said output of said logic element.

24. The configurable electronic device according to claim 23 wherein a plurality of said logic elements in one of said logic blocks each have outputs selectively connectable to a wide-input AND gate.

25. The configurable electronic device according to claim 23 wherein said at least one logic element further comprises:

a plurality of input multiplexers for selectively connecting said logic element inputs and said shared inputs to inputs of said look-up table; and

a plurality of output multiplexers for selectively connecting outputs of said look-up table to said logic element outputs.

26. The configurable electronic device according to claim 23 wherein said at least one logic element has an output which may be programmably connected to its inputs to perform any one of the functions: H(a,b,c,d); F(y,b,c)&d # G(a,b,y)&!d; F(a,b,c)&x # G(a,d,c)&!x; or F(a,b,y)&x # G(c,d,y)&!x where a, b, c, and d represent non-shared inputs of said logic element, x and y represent shared inputs connected to a plurality of logic elements, & represents a logical AND operation, ! represents a negation, and # represents a logical OR operation.

27. A configurable logic device comprising:

a plurality of logic array blocks (LABs), arranged in a plurality of rows and columns, at least one LAB having a plurality of inputs and outputs and capable of performing a logic function such that said at least one LAB's outputs represent a logical function of its inputs, said logic function being programmable during chip manufacture or use;

a general interconnect structure comprising a plurality of rows of conductors and a plurality of columns of conductors, said interconnect structure capable of routing select signals between said columns of conductors and said rows of conductors;

LAB output lines for selectively connecting said LAB outputs to said general interconnect structure;

at least one LAB-based interconnect associated with a particular LAB capable of connecting selected lines of said general interconnect to selected input lines of said LAB;

at least one shared LAB-wide input line;

wherein at least one LAB is comprised of a plurality of logic elements each of said logic elements having at least one output and a plurality of inputs and each of said logic elements performing a function such that its outputs are a logical function of its inputs said logic elements comprising:

at least two look-up tables, said look-up tables each having an output and a plurality of inputs;

a plurality of input multiplexers for selectively connecting said plurality of logic element inputs to said look-up table inputs; and

a plurality of output multiplexers for selectively connecting one output of said plurality of look-up tables to an output of said logic element.

28. The configurable electronic device according to claim 27 wherein a plurality of said logic elements are four input logic elements and wherein a plurality of said look-up tables are three input look-up tables and wherein there are two shared LAB input lines.

29. The configurable logic device of claim 27 wherein shared input lines may be selectively connected to a line in said local interconnect.

30. A logic element for use in a programmable logic device comprising:

a plurality of unshared input signal lines, said input signals not shared with other logic elements;

a plurality of shared input signal lines, said shared input signal lines shared with other logic elements;

first and second look-up tables each having an output and a plurality of inputs and each being programmable such that its output can represent a logical function of its input;

a first input multiplexer with its output coupled to an input of said first look-up table, one input coupled to one of said plurality of unshared input signal lines and a second input coupled to one of said plurality of shared input signal lines;

a second input multiplexer with its output coupled to an input of said first look-up table, one input coupled to one of said plurality of unshared input signal lines and a

second input coupled to another of said plurality of unshared input signal lines;

a third input multiplexer with its output coupled to an input of said second look-up table, one input coupled to one of said plurality of unshared input signal lines and a second input coupled to one of said plurality of shared input signal lines; and

an output multiplexer with its output coupled to an output of said logic element, one input coupled to an output of said first look-up table and a second input coupled an output of said second look-up table.

31. The logic element according to claim 30 further comprising a fourth input multiplexer with its output coupled a select signal of said output multiplexer, one input coupled to one of said plurality of unshared input signal lines and a second input coupled to one of said plurality of shared input signal lines.

32. The logic element according to claim 30 further comprising a first programmable store coupled to the select signal of said first input multiplexer and said third input multiplexer.

33. The logic element according to claim 32 further comprising a second programmable store coupled to the select signal of said third input multiplexer and said fourth input multiplexer.

34. A configurable electronic device comprising:

a plurality of logic elements grouped into a plurality of logic blocks, at least one logic element comprising a plurality of outputs and a plurality of inputs wherein said outputs are a logical function of said inputs and wherein at least one logic block has associated with it:

a wide-input AND gate, said wide-input AND gate having inputs connectable to a plurality of outputs of said

plurality of logic elements and having an output that is a logical function of its inputs.

35. The configurable electronic device according to claim 34 wherein at least one of said logic element outputs is selectively connectable to said wide-input AND gate.

36. The configurable electronic device according to claim 34 whherein at least one of said logic element outputs may be selectively inverted prior to connection to said wide-input AND gate.

37. The configurable electronic device according to claim 36 wherein at least one of said logic element outputs is selectively connectable to said wide-input AND gate by means of an OR gate with its output connected to said wide-input AND gate, one input connected to the output of said at least one logic element and a second input connected to a programmable store wherein said programmable store may be programmably set to either a true or false logic value.

38. The configurable electronic device according to claim 37 wherein at least one of said logic element outputs may be selectively inverted prior to connection to said wide-input AND gate by means of an XOR gate with its output connectable to said wide-input AND gate, one input connected to the output of said at least one logic element and a second input connected to a programmable store wherein said programmable store may be programmably set to either a true or false logic value.

39. A configurable logic device comprising:

a plurality of logic array blocks (LABs), arranged in a plurality of rows and columns, at least one LAB having a plurality of inputs and outputs and capable of performing a logic function such that said at least one LAB's outputs represent a logical function of its inputs, said logic function being programmable during chip manufacture or use;

a general interconnect structure comprising a plurality of rows of conductors and a plurality of columns of conductors, said interconnect structure capable of routing select signals between said columns of conductors and said rows of conductors;

LAB output lines for selectively connecting said LAB outputs to said general interconnect structure;

at least one LAB-based interconnect associated with a particular LAB capable of connecting selected lines of said general interconnect to selected input lines of said LAB;

wherein at least one LAB is comprised of a plurality of logic elements, each of said logic elements having at least one output and a plurality of inputs and each of said logic elements performing a function such that its outputs are a logical function of its inputs; and

at least one wide-input AND gate, said wide-input AND gate having inputs connectable to a plurality of outputs of said plurality of logic elements and having an output that is a combinatorial logical function of its inputs.

40. The configurable electronic device according to claim 39 wherein at least one of said logic element outputs is selectively connectable to said wide-input AND gate by means of an OR gate with its output connected to said wide-input AND gate, one input connected to the output of said at least one logic element and a second input connected to a programmable store wherein said programmable store may be programmably set to either a true or false logic value.

41. The configurable electronic device according to claim 39 wherein at least one of said logic element outputs may be selectively inverted prior to connection to said wide-input AND gate by means of an XOR gate with its output connectable to said wide-input AND gate, one input connected to the output of said at least one logic element and a second input connected to a programmable store wherein said programmable store may be programmably set to either a true or false logic value.

42. The configurable logic device of claim 39 wherein said at least one LAB-based input interconnect is a fully populated interconnection between inputs to logic circuits of said LAB and outputs of a plurality of multiplexers, said multiplexers having their inputs connected to conductors in said general interconnect.

43. The configurable logic device of claim 39 wherein an output of said wide-input AND gate may be selectively connected to a line in said global interconnect.

44. The configurable logic device of claim 39 wherein the output of said wide-input AND gate is connectable to said general interconnect structure.

45. The configurable logic device of claim 39 wherein at least one input of said wide-input AND gate is connectable to said at least one LAB-based interconnect.

46. The apparatus according to any of claims 8 to 45 wherein at least one logic element further has associated with it a cascade selector with its output coupled to an input of said logic element, one of its selectable inputs coupled to said cascade connector and another of its selectable inputs coupled to a signal providing a constant logic value.

47. The apparatus according to any of claims 1 to 45 including at least one logic element comprised of a plurality of look-up tables.

48. The apparatus according to claim 47 wherein data in said plurality of look-up tables is stored in random access memory.

49. The apparatus according to any of claims 1 to 45 further comprising a plurality of multiplexers with inputs connected to said general interconnect structure and outputs connected to input interconnections.

50. The configurable electronic device as recited in claim 49 wherein said general interconnect structure comprises an array of vertical and horizontal interconnect lines between logic blocks.

51. The configurable electronic device as recited in claim 49 further comprising an array of local interconnection lines that is fully populated to inputs of individual logic elements in said logic block.

52. The apparatus as recited in any one of claims claims 1 to 51 wherein the apparatus is connected in a computing system, said system comprising:

a processing unit;

a memory; and

an input/output interface.

53. A method in a programmable logic device of logically combining logic element signals comprising the steps of:

providing at least one logic element with a cascade logic gate;

routing an output signal from another logic element to an input of said cascade logic gate such that said output signal may be selectably combined with a signal directed to said input.

54. A method in a programmable logic device of logically combining logic element signals comprising the steps of:

providing at least one logic element with a cascade logic gate;

routing an output signal from another logic element to an input of said cascade logic gate such that said output signal may be selectably combined with a signal directed to said input.

55. A method in a programmable logic device of routing signals in the programmable logic device to device output pins comprising the steps of:

providing a plurality of LABs with a LAB-based interconnect for connecting signals on the general interconnect to input signals in the LAB; and

including in a plurality of LAB-based interconnects a plurality of output lines, said lines being selectively connectable to the same general interconnect lines as are the LAB inputs and said output lines each being connectable to an output pin.

56. A method in a programmable logic device of enhancing functionality of logic elements comprising the steps of:

grouping said logic elements into a plurality of logic blocks;

constructing for at least one logic block a plurality of shared input signal lines for providing shared input signals to a plurality of logic elements;

generating an output of at least one of said logic elements in said at least one logic block through use of first and second look-up tables, said look-up tables having inputs selectively connectable to unshared inputs of said logic element and to said shared input signal lines of said logic block; and

providing a means for selectively connected outputs of said logic elements to outputs of said look-up tables.

57. A method in a programmable logic device of logically combining logic element signals comprising the steps of:

grouping said logic elements into a plurality of logic blocks;

providing at least one logic block with a wide-input AND gate; and

providing a means for selectively connected outputs of logic elements grouped into said at least one logic block to inputs of said wide-input AND gate.

58. A logic array block for a programmable logic device comprising:

a first dedicated input;

a first programmable function block for implementing logic functions;

a second programmable function block, programmably coupled to said dedicated input and said first programmable function block, said second programmable function block for implementing logic functions; and

a local interconnect region, programmably coupling an output of said first programmable function block and an output of said second programmable function block to inputs of said logic array block, without passing through a global interconnect region of said programmable logic device.

59. The logic array block of claim 58 further comprising a third programmable function block, coupled to said first programmable function block, said third programmable function block for implementing logic functions, wherein an output of said third programmable function block is programmably coupled through said local interconnect structure to said inputs of said logic array block, without passing through said global interconnect region of said programmable logic device.

60. The logic array block of claim 59 further comprising:

a second dedicated input;

a fourth programmable function block, coupled to said third programmable function block, and programmably coupled to said second dedicated input and said second programmable function block, wherein an output of said fourth programmable function block is programmably coupled through said local interconnect structure to said inputs of said logic array block, without passing through said global interconnect region of said programmable logic device.

61. The logic array block of claim 58 further comprising:

a storage block, programmably coupled to said first dedicated input and said first programmable function block.

62. The logic array block of claim 59 further comprising:

a storage block, programmably coupled to said first dedicated input, said first programmable function block, and said third programmable function block.

63. The logic array block of claim 61 wherein an output of said storage block is not programmably coupled to said local interconnect structure.

64. The logic array block of claim 58 wherein said first programmable function block and said second programmable function block are substantially identical.

65. The logic array block of claim 59 wherein said first programmable function block and said second programmable function block are four-input look-up tables and said third programmable function block is a two-input look-up table.

66. The logic array block of claim 59 wherein said programmable function blocks are implemented in a static random access memory.

67. A system incorporating a programmable logic device with a logic array block as recited in claim 58.

68. A logic array block for a programmable logic device comprising:

a first primary four-input function block for implementing logic functions;

a second primary four-input function block for implementing logic functions;

a third primary four-input function block for implementing logic functions;

a fourth primary four-input function block for implementing logic functions; and

a secondary four-input function block, for implementing logic functions, wherein a first input of said

secondary four-input function block is programmably coupled to said first primary four-input function block and a first dedicated input line, a second input of said secondary four-input function block is programmably coupled to said second primary four-input function block and a second dedicated input line, a third input of said secondary four-input function block is programmably coupled to said third primary four-input function block and a third dedicated input line, and a fourth input of said secondary four-input function block is programmably coupled to said fourth primary four-input function block and a fourth dedicated input line.

69. The logic array block of claim 68 further comprising:

a first secondary two-input function block, wherein a first input of said first secondary two-input function block is coupled to said first primary four-input block, and a second input of said first secondary two-input function block is coupled to said second primary four-input block; and

a second secondary two-input function block, wherein a first input of said second secondary two-input function block is coupled to said third primary four-input block, and a second input of said second secondary two-input function block is coupled to said fourth primary four-input block.

70. The logic array block of claim 69 further comprising:

a first register, programmably coupled to said second dedicated input line, said second primary four-input function block, and said secondary four-input function block; and

a second register, programmably coupled to said secondary four-input function block, said third primary four-input function block, and said third dedicated input line.

71. The logic array block of claim 70 further comprising:

a third register, programmably coupled to said first dedicated input line, said first secondary two-input function block, and said first primary four-input function block; and

a fourth register, programmably coupled to said fourth primary four-input function block, said second secondary two-input function block, or said fourth dedicated input line.

72. The logic array block of claim 70 wherein said register further comprises:

a clock input, selectively coupled to a first clock source or a second clock source;

a clock enable control, for enabling clocking by said clock input of said register; and

a set and reset control, for setting and resetting said register.

73. A programmable logic array integrated circuit organized as a two-dimensional array of cells comprising:

a first plurality of conductors extending along a first dimension of said two-dimensional array;

a second plurality of conductors extending along a second dimension of said two-dimensional array, said second plurality of conductors programmable coupled to said first plurality of conductors; and

a plurality of logic array blocks, wherein a logic array block comprises:

a plurality of first-level programmable function blocks for implementing logical functions;

a plurality of second-level programmable function blocks, programmably coupled to said plurality of first-level programmable function blocks, without passing through said first plurality of conductors and said second plurality of conductors, said plurality of second-level programmable function blocks for implementing logical functions of outputs from said plurality of first-level programmable function blocks; and

a plurality of dedicated inputs programmably coupled to said plurality of second-level programmable function blocks.

74. The programmable logic array of claim 73 further comprising a plurality of switch boxes for programmably coupling said first plurality of conductors to said second plurality of conductors.

75. The programmable logic array of claim 73 further comprising a plurality of switch boxes for programmably coupling said first plurality of conductors to other conductors within said first plurality of conductors.

76. The programmable logic array of claim 75 wherein said first plurality of conductors programmable couples signals between at least two logic array blocks without passing through said plurality of switch boxes.

77. The programmable logic array of claim 73 further comprising a plurality of multiplexer regions for coupling said first plurality of conductors and said second plurality of conductors to said logic array blocks.

78. The programmable logic array of claim 73 further comprising a first plurality of long conductors, extending in said first dimension of said two-dimensional array, said first plurality of long conductors for programmably coupling a plurality of logic array blocks.

79. The programmable logic array of claim 78 further comprising a plurality of multiplexer regions for programmably coupling said first plurality of long conductors to said second plurality of conductors.

80. The programmable logic array of claim 73 further comprising a first plurality of long conductors, extending in said first dimension of said two-dimensional array, said first

plurality of long conductors programmably coupling said plurality of logic array blocks, wherein said plurality of logic array block not adjacent to another.

81. The programmable logic array of claim 80 wherein outputs from said logic array blocks are programmably coupled directly to said first plurality of long conductors, without passing through a multiplexer region.

82. The programmable logic array of claim 80 wherein outputs from said logic array blocks are programmably coupled directly to said first plurality of long conductors, and said first plurality of long conductors are programmably coupled to inputs of said logic array blocks through a multiplexer region.

83. The programmable logic array of claim 78 wherein said first plurality of long conductors forms a tristateable bus.

84. The programmable logic array of claim 73 further comprising a plurality of double-length lines for programmably coupling two logic array blocks without using a switch box.

85. The programmable logic array of claim 73 wherein said first plurality of conductors and said second plurality of conductors comprise double-length lines.

86. The logic array block of claim 58 further comprising:

a carry logic block, coupled to said first programmable function block, said carry logic block generates or propagates a carry signal in response to inputs to said first programmable function block.

87. The logic array block of claim 58 wherein said first programmable function block is configurable as a random access memory for storing data, wherein address data is provided for

said random access memory through inputs of said first programmable function block.

88. The logic array block of claim 87 further comprising a program register for synchronizing a write enable signal with a clock source.

89. The logic array block of claim 87 further comprising a write enable logic block, coupled to said random access memory, said write enable logic block is responsive to a write enable signal to enable reading and writing of said random access memory.

76

Patents Act 1977 Examiner's report to the Comptroller under Section 17 (The Search report)	Application number GB 9522616.3
Relevant Technical Fields (i) UK Cl (Ed.O) G4H (HU), H3P (PHX) (ii) Int Cl (Ed.6) H03K (19/173, 19/177)	Search Examiner MR K SYLVAN Date of completion of Search 26 JANUARY 1996
Databases (see below) (i) UK Patent Office collections of GB, EP, WO and US patent specifications. (ii)	Documents considered relevant following a search in respect of Claims :- 1-7, 53, 54

Categories of documents

X:	Document indicating lack of novelty or of inventive step.	**P:**	Document published on or after the declared priority date but before the filing date of the present application.
Y:	Document indicating lack of inventive step if combined with one or more other documents of the same category.	**E:**	Patent document published on or after, but with priority date earlier than, the filing date of the present application.
A:	Document indicating technological background and/or state of the art.	**&:**	Member of the same patent family; corresponding document.

Category	Identity of document and relevant passages		Relevant to claim(s)
A	WO 94/17595 A1	(XILINX) see Abstract	
X	US 5350954	(ALTERA) see Figures and column 2, lines 41 to 45	3, 53, 54 at least
X	US 5220214	(ALTERA) see the Figure	3, 53, 54 at least

Databases:The UK Patent Office database comprises classified collections of GB, EP, WO and US patent specifications as outlined periodically in the Official Journal (Patents). The on-line databases considered for search are also listed periodically in the Official Journal (Patents).